UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

MATHEMATICAL THEORY IN CLASSIFICATION AND SEGMENTATION

A DISSERTATION

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

Degree of

DOCTOR OF PHILOSOPHY

By

KESONG CHENG
Norman, Oklahoma
2012

MATHEMATICAL THEORY IN CLASSIFICATION AND SEGMENTATION

A DISSERTATION APPROVED FOR THE
DEPARTMENT OF MATHEMATICS

BY

Dr. Meijun Zhu, Chair

Dr. Yang Hong

Dr. Shihshu Walter Wei

Dr. Nikola Petrov

Dr. Luther White

DEDICATION
to

The Love of My Life
Xueqi Zhou

ACKNOWLEDGEMENTS

In the past six years, my advisor Professor Meijun Zhu has been guiding and teaching me both academically and personally. I would like to express my greatest gratitude to him. I would like to thank Professor Luther White, Professor Shihshu Walter Wei, Professor Nikola Petrov and Professor Yang Hong for serving my committee and helping me throughout the journey. The Department of Mathematics at the University of Oklahoma kindly provided me with financial assistance during the study. I also want to extend my appreciation to my colleagues Weihua Lin, Ye Li, Shiyun Tang, my parents, and my wife Xueqi Zhou for their support and encouragement.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

This thesis, consisting of three chapters, develops and adapts several mathematical approaches in solving a series of questions of interdisciplinary research interests.

In the first chapter, we design a new metric function for the verification of meteorological forecasts. The core question is to compare various meteorological forecasts with the observation. To solve the problem, we propose a metric function to evaluate the difference between forecasts and observations. Moreover, a comparison method is introduced to compare across different forecasts and selects the optimal predictor that best matches with observations.

In the second chapter, we create a nowcast of rotating storms in real-time based on the global properties of the storms. Advanced active contour method is utilized on storm images to capture unique characteristics of storms and novel modeling of rotating storm motion is employed to simulate movements of the storm. Real time images are used to verify that the prediction of the efficient forecast lines up with the actual route.

In the third chapter, we develop an algorithm in data clustering and apply it to construct hierarchical diagrams. Using a newly devised algorithm, a set of points could be represented by just a single point. This algorithm is implemented in the construction of a new hierarchical tree. Computer-based experiments are conducted to demonstrate the difference between the new hierarchical clustering method and the previous method.

CHAPTER 1
HAUSDORFF METRIC IN SPATIAL VERIFICATION

In this chapter we introduce a new method to verify meteorological forecasts. The new method includes a metric function which provides evaluations of different forecasts. It is more resistant to perturbations and addresses specific problems raised by previous verification methods. We also discuss another scalar function, which does not satisfy positive property of a metric function but fulfills the other two properties. It is also applied to compare different forecast with the observation.

1.1 Introduction

The verification of high-resolution forecasts tend to have more significant position errors than the verification of short time high-resolution forecasts. The position errors could be so high that there is no overlap at all between observations and forecasts. Therefore, a method beyond simple pixel-to-pixel correspondence is needed to reward a forecast which has some correct prospects. In the absence of such a reward, the verification method will suffer from a "double-penalty" problem (Gilleland et al., 2009; Ahijevych et al., 2009).

To solve the problem of comparing forecasts with observations when significant displacement errors appear in the forecasts, several verification techniques have been introduced. Instead of computing errors by directly differencing two fields, in neighborhood approaches, a neighborhood around each grid point is searched in both fields and the statistical properties of the set of pixels in the neighborhoods such as mean value are compared. Pixel-to-pixel correspondence requirements can also be

avoided by comparing properties of the pixels in the entire domain of interest, as was done by Wernli et al. (2009). The resolution at which double penalty errors start to show up is an indirect measure of the position error although the errors may well not be due to location.

One way to categorize these methods is to consider their purpose. They all intend to modify the image or the range of pixels so that pixel-to-pixel (or super-pixels to super-pixels) error measurements work. Despite the richness of the verification measures, it has been known that simple and intuitive scalar measures of performance are more efficient and easier to use. Our proposal, then, is to devise an intuitive scalar measure of model performance that can be computed without extensive preprocessing of model forecast fields. A distance metric is certainly intuitive — the "farther" away a forecast is from the observation field, the worse it is. It is also a scalar and has the benefit of naturally encompassing position errors in model forecasts. In this chapter we will devise a distance metric that can be used to gauge how close a forecast is to the observation. It should be noted that the metric introduced in this chapter is a distance, not a skill score. Unlike a skill score, the distance is not bounded — the larger the distance, the worse the forecast.

It is important that verification measurements are metrics because in the absence of it being a metric, we may obtain unreasonable results when comparing two forecasts. The positivity property notes specifically that the distance between two objects is zero is equivalent to the fact that these two objects are identical. This is important because in a perfect forecast, the sets of pixels corresponding to the

observation and forecast fields will be identical and it is necessary to recognize a perfect forecast.

The triangle inequality property is essential to carry out a fair measurement. For example, in the following scenario like this: Let O be the observation, F1 and F2 be two forecasts. If we measured that the distance between O and F1 is 100 units, and the distance between O and F2 is 10 units, we would say that F2 is a better forecast. However, if the verification measurement does not satisfy the triangle inequality property, we may find that the distance between F1 and F2 is 0.5 units or even less. Considering the expected variance in computed distances, we may not be convinced that F2 is really better since it is almost the same as F1 (the distance between them is almost zero).

The symmetric property guarantees that every set has equal right to be fairly measured: the distance from set A to set B is always the same as the distance from set B to set A.

1.2 A metric between two sets

Although the definition of a metric seems intuitive, many reasonable measurements turn out to not be metrics especially when considering the model verification problem. This is because it is non-trivial to define a distance between two sets of points that is a metric. Even though defining a metric between a point and a set of points is not difficult, it is a different situation when dealing with two sets of points.

1.2.1 Distance between a point and a set

We take a look at the most intuitive measurement, the Euclidean metric. For two points $x = (x_1, x_2)$, $y = (y_1, y_2) \in \mathbb{R}^2$, the Euclidean metric function is $m(x, y)$ defined by:

$$m(x, y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}. \quad (1.1)$$

This is shown in Figure 2.1b. Given the Euclidean metric between two points, we can define another metric, this time between any point $x \in \mathbb{R}^2$,and a set $A \subset \mathbb{R}^2$ as:

$$m(x, A) = \min_{y \in A} m(x, y) \quad (1.2)$$

i.e. as the distance between the point x and the closest point to it in the set A (See Figure 1.1a). It can be noted that metric $m(x, A)$ is overly sensitive. It is defined exclusively by the closest point and can therefore be unduly affected by noise in the data. Consider the scenario in Figure 1.1b where the forecast field has a single non-zero pixel close to the observations. Because this pixel is closest to the observations, all the $m(x, A)$ will be evaluated on the basis of this one point. We will look at distances between two sets of points in the next section.

1.2.2 Hausdorff distance between two sets

In this section, we use the distance between a set of points and a single point to define the distance between two sets of points, which would be needed to find the distance between the pixels of the forecast and observed images. One possibility is to define it as the intuitively appealing maximum of all possible $m(x, A)$:

$$d(X, A) = \max_{x \in X} m(x, A) \tag{1.3}$$

This Euclidean distance between two sets turns out to not even be a metric function as it is not symmetric. In other words, $d(X, A)$ can be different from $d(A, X)$ (See Figure 1.1c for an illustration). Using the minimum of all possible $m(x, A)$ does not work either because it does not satisfy the positivity property. The sets A and X need to only overlap, not be identical, for the distance to be zero.

Hausdorff metric gives a solution to this problem by applying the symmetric method:

$$m_H(A, B) = \max\{d(A, B), d(B, A)\}. \tag{1.4}$$

The mathematical definition of a metric space and Hausdorff metric is shown in the following:

Definition: A metric space is a set M in which a distance function d is defined, with the following properties:

(a) $0 \leq d(x,y) < \infty \ \ for\ all\ x\ and\ y \in M.$ (b) $d(x,y) = 0\ if\ \ and\ \ only\ \ if\ \ x = y.$

(c) $d(x,y) = d(y,x)\ for\ all\ x\ and\ y \in M.$

(d) $d(x,y) \leq d(x,z) + d(z,y) for\ all\ x,\ \ y\ \ and\ z \in M.$

Definition: Let $A\ and\ B$ be two non-empty subsets of a metric space (M,d). The Hausdorff distance between $A\ and\ B$ is defined by:

$$m_H(A,B) = \max\left\{\sup_{x\epsilon A}\inf_{y\in B} d(x,y)\,, \sup_{y\epsilon B}\inf_{x\in A} d(x,y)\right\}.$$

Since the sets we are discussing about are closed sets, the maximum and minimum could be used to substitute for sup and inf.

It should be noted that when B includes only one point x, the distance from x to set A is different from the Hausdorff distance between x and set A: the former measures the point x to the closest point to it in set A, the latter measures x to the farthest point to it in set A. The maximum operation in the Hausdorff metric makes it very susceptible to noise. One possible way to address this, called the Partial Hausdorff Distance, is to use a percentile to replace the maximum. However, this is not a metric anymore, so most methods that are based on the Partial Hausdorff Distance are not metrics either. Baddeley (1992) replaced the maximum in the definition of the Hausdorff metric with an L^p norm and this was employed for model forecast

verification by Gilleland et al. (2008). Similar to the Hausdorff metric, such a metric may suffice when the objective is to compare objects that consist of contiguous sets of pixels i.e. if there will not be noisy pixels elsewhere in the image that have to be considered part of the distance computation. If these Hausdorff type metrics are not preceded by a step of object identification or noise removal, they are always sensitive to noise. This is because of the property of Euclidean metric function $m(x, A)$, and Hausdorff type metrics are built on is itself problematic for spatial field verification, as opposed to verifying objects extracted from those fields.

1.3 Verification metric

In this section, we introduce an easily computable metric that has been devised specifically for the model verification problem. Rather than consider a generic pair of binary images, we recognize that, in model verification, there is an observation field which is quite special and a set of forecast fields each of which has to be evaluated. Our metric will use the observation field as a reference field so as to come up with a measure that is (a) a metric, and so can be used to rank forecasts, (b) able to evaluate between two sets and does not require pixel-to-pixel correspondence.

The metric can be computed directly from the images. It is not necessary to filter, warp, window, identify objects or fit parameters to the images. It should be noted that our metric is defined on sets of pixels and, so, it requires a threshold to be specified. Pixels with a data value greater than the threshold will be considered part of the set and those with a data value less than the threshold will be considered outside the set, thus images are converted into binary images first .When we show the results of our technique ,we will demonstrate the results on a variety of thresholds.

Let O be the set consisting of pixels in the observation field that are above a specified threshold. Let A, B be sets consisting of pixels in forecast fields. The verification metric that we propose is as follows.

1.3.1 Definition of verification metric

Definition: The verification metric between two sets A and B is defined by

$$metr_v(A,B) \coloneqq \lambda_1 dist_{OV}(A,B) + \lambda_2 dist_{DV}(A,B) \tag{1.5}$$

i.e. a weighted sum of two distances that are defined below. The overlap-based distance $dist_{OV}$ is defined by:

$$dist_{OV}(A,B) = \sqrt{\sum_{\forall i}\sum_{\forall j}(a_{ij}-b_{ij})^2} \tag{1.6}$$

where a_{ij}, b_{ij} are characteristic functions of sets A, B, respectively. i.e. a_{ij} is 1 if the pixel (i,j) is in the set A, a_{ij} is 0 if the pixel $(i,j)$is not in the set A. b_{ij} is defined similarly.

Next, we introduce the observation distance. The observation distance $dist_{ob}$ is the average distance of every observation point to a forecast field:

$$dist_{ob}(O,A) = \begin{cases} \dfrac{1}{N(O)} \displaystyle\sum_{i=1}^{N(O)} m(o_i, A) & if \quad N(O) \cdot N(A) \neq 0 \\ 0 \quad if \quad N(O) = 0 \ and \quad N(A) = 0 \\ D \quad otherwise \end{cases} \tag{1.7}$$

where $m(o_i, A)$ is the Euclidean metric function of Equation (1.2), o_i are the pixels in the observation field, $N(O)$, $N(A)$ are the number of pixels in the sets O and A, respectively, i.e. the number of pixels in the corresponding images that are above the threshold. The number D in the definition of $dist_{ob}$ is a number larger than the maximum possible distance. One possible choice is the length of the diagonal of the grids being compared. This upper limit value of $dist_{ob}$ will be reached if the observation field or the forecast field is an empty set.

The above distance is used to compute the observation based displacement $dist_{DV}$ between the sets A and B as:

$$dist_{DV}(A,B) = |dist_{ob}(O,A) - dist_{ob}(O,B)|. \tag{1.8}$$

The relative weights of the two component distances (of Equation (1.6) and (1.8)) in the verification metric are quite subjective. We use $\lambda_1 = \lambda_2 = \frac{1}{2}$ throughout this chapter for simplicity.

Different weights could be chosen depending on whether overlap error is more or less important than displacement. As defined, the units of the measurement are in pixels. It can be converted into a real world distance by multiplying by the appropriate pixel dimensions.

1.3.2 Simplified form of verification metric

Since this metric will mainly be used for verification, one of the terms in $dist_{DV}$ drops away (since $dist_{ob}(O, O) = 0$), leaving:

$$metr_V(O, A): \\ = \lambda_1 \, dist_{OV}(O, A) + \lambda_2 dist_{ob}(O, A). \qquad (1.9)$$

Other than to prove the triangle inequality, when we will need the more general form, this simplified definition with $\lambda_1 = \lambda_2 = \frac{1}{2}$ is what we will term the verification metric.

The observation file O plays a special role in $dist_{DV}$ (See Equation (1.8)). The distance between any two fields A and B is computed as the sum of the distances between each of those fields and O. In other words, the observation field is the reference field against which forecasts are compared as far as their displacement is compared. The overlap between forecasts, on the other hand, is compared directly from the two fields (See Equation (1.6)). If we are comparing a forecast field to an observation field, one of the terms in $dist_{DV}$ is zero and both comparisons take place on an image-to-image level.

Further $dist_{DV}$ does not penalize overforecasts. For example, consider the scenario in Figure 1.1c. For every point in the observation, there is a point in the forecast field that exactly matches. Therefore, $dist_{Ob}$ is zero, leading to a zero $dist_{DV}$. Thus, one way of thinking about the overlap term $dist_{OV}$ is as the penalty for over

forecasts. On the other hand, $dist_{OV}$ is based on strict pixel-to-pixel correspondence and is, therefore, insensitive to position errors — the scenarios in Figure 1.1 d and e have the same $dist_{OV}$ but the $dist_{DV}$ of Figure 1.1 e is larger, leading to a larger value in the verification metric. In this view, $dist_{DV}$ provides the position-error sensitivity to the verification metric.

It should be noted that the verification metric is a distance and not a bounded skill score. The larger the images being compared, the larger the maximum distance can be. The images being compared should be of the same size and resolution. In practice, this can be achieved by cropping or sub sampling the larger or more detailed image to meet the dimensions and resolution of the smaller, coarser image.

It should also be noted that the metric is extremely sensitive to the observation field, because distances are defined by using the observation as the reference field. This is because the verification metric is designed to compare two forecasts given the same observation. The verification metric should not be used to compare two forecasts at two different times — a forecast 100 km displaced from the observation might be acceptable when there are only a few observations, but may not be acceptable when the entire domain is full of observations.

1.4 Comparison method

It is worth mentioning another method of calculating the distance between two sets of points. Even though this method does not satisfy the positive property of a metric, it does satisfy the symmetric property and triangle inequality. Thus, when we

have multiple forecasts, this method could be used to calculate their distances to the observations and the results could be compared to find the best forecast.

Definition: For two given sets A and B in a metric space (M, d), suppose the points in them are listed as $a_1, a_2, a_3, \dots, a_i, \dots, a_m \in A, \quad b_1, b_2, b_3, \dots, b_j, \dots, b_n \in B,$ the comparison distance between A and B is defined as following:

$$\mathrm{dist}_{\mathrm{comp}}(\mathrm{A}, \mathrm{B}) = \frac{1}{mn} \cdot \sum_{i=1}^{i=m} \sum_{j=1}^{j=n} d(a_i, b_j) \tag{1.10}$$

It should be noted that when using the comparison method on two sets, both of the sets should be on the same field, which means they have the same coordinate system. This is because we rely on the coordinates of the points to calculate the distance between points.

When A and B are the same set with more than one point, $d(a_i, b_j)$ is not necessarily equal to zero because there are different points in the sets. This will lead to a positive value of $\mathrm{dist}_{\mathrm{comp}}(\mathrm{A}, \mathrm{B})$, which means the distance between two identical sets is greater than zero. Even though, the comparison distance does not satisfy the positive property of a metric, it satisfies the trigonometric inequality and the symmetric property as shown in the following proof.

Proof: Given three sets A, B and C and their corresponded points $a_1, a_2, a_3, \dots, a_i, \dots, a_m \in A, \quad b_1, b_2, b_3, \dots, b_j, \dots, b_n \in B, \quad c_1, c_2, c_3, \dots, c_k, \dots, c_l \in C.$ Since the distance between points is a metric, it satisfies the trigonometric inequality:

$$d(a_i, b_j) + d(b_j, c_k) \geq d(a_i, c_k)$$

where $a_i \in A$, $b_j \in B$, $c_k \in C$, $i = 1,2,\ldots,m$, $j = 1,2,\ldots,n$, $k = 1,2,\ldots,l$.

By taking a summation on both sides with respect to i, j and k, we have:

$$\sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{l} d(a_i, b_j) + \sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{l} d(b_j, c_k) \geq \sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{l} d(a_i, c_k)$$

which is the same as:

$$l \cdot \sum_{i=1}^{m}\sum_{j=1}^{n} d(a_i, b_j) + m \cdot \sum_{j=1}^{n}\sum_{k=1}^{l} d(b_j, c_k) \geq n \cdot \sum_{i=1}^{m}\sum_{k=1}^{l} d(a_i, c_k)$$

dividing $l \cdot m \cdot n$ on both sides of the inequality, we have:

$$\frac{1}{mn} \cdot \sum_{i=1}^{m}\sum_{j=1}^{n} d(a_i, b_j) + \frac{1}{nl} \cdot \sum_{j=1}^{n}\sum_{k=1}^{l} d(b_j, c_k) \geq \frac{1}{ml} \cdot \sum_{i=1}^{m}\sum_{k=1}^{l} d(a_i, c_k)$$

which is equivalent to:

$$d(a_i, b_j) + d(b_j, c_k) \geq d(a_i, c_k)$$

The symmetric property of comparison distance could be derived instantly from the symmetric property of Euclidean distance.

■

Since the comparison method satisfies the trigonometric inequality, it could be used to compare the distances between different forecasts and the observation, the best forecast would have the smallest distance using comparison method. In the mean time,

the comparison method uses the average of all the distances between every possible different pairs of points, it decreases the influence of perturbations and noises and supplies a relatively stable algorithm to calculate the distance between forecasts and observations.

1.5 Experiments

We computed the verification metric and comparison distance on a geometric and on a perturbed dataset from a verification method inter-comparison project (Gilleland et al., 2009; Ahijevych et al., 2009) that was established to improve the understanding of the characteristics of various spatial forecast verification methods. To enable reasonable inter-comparison, the verification methods were carried out on synthetic and real fields with known errors. The methods were also applied to real model forecasts from an experiment conducted by Kain et al. (2008). The results of the verification metric and comparison distance on the different datasets that were created by the inter-comparison project are presented below.

1.5.1 Geometric cases

The "geometric" were defined on a 601×501 grid and were mapped to a 601×501 subsection of the NCEP storage grid 240. The geometric cases illustrate three types of error: 1) displacement, 2) frequency, and 3) aspect ratio. The images are shown in Figure 1.2. The description of the results are given in Table 1.1.

Because both verification metric and comparison distance are defined on binary images, the fields are filtered with threshold at zero i.e. pixels with a value above zero are assumed to be part of the object and pixels in the "white" background are assumed to be outside it. In particular, this means that even though the objects have two intensity levels, they are treated as a single intensity level.

The steps to compute the verification metric for forecast1 are delineated in the following. First, the observation field and forecast field (forecast1) are both under threshold of zero. Thus, there are two binary images. The first image consists of pixels whose value is 1 within the ellipse of observation and 0 outside. The second image is similar, except that the ellipse corresponds to the points in forecast1. From these two binary images, the verification metric needs to be computed using Equation (1.9). The second step, then, is to compute $dist_{OV}$, defined in Equation (1.6). a_{ij} is 1 within the first ellipse while b_{ij} is 1 within the second ellipse. If the ellipses had overlapped, the difference $a_{ij}-b_{ij}$ would have been zero at points of overlap. Here, however, the ellipses do not overlap. Thus, the difference has a magnitude of 1 where either a_{ij} or b_{ij} is 1. Therefore, $dist_{OV}$ is equal to the square root of twice the size of the ellipse measured in pixels. The third step is to compute $dist_{ob}$ using Equation (1.7). Both the observation and the forecast have some valid points, so the answer is not simply the length of the diagonal of the grids being compared. Instead, the Euclidean distance from every observation point to the closest point in the forecast field needs to be computed. For every point within the ellipse in the observation, we need to find the closest point in forecast1. It should be noted that we will find the closest point, not the corresponding point. Because forecast1 consists of the ellipse displaced right, the closest points will

all consist of points on the leftmost boundary of the ellipse in forecast1. For the points on the left boundary of the ellipse in the observation, m_{OA} will be 50 units, the known displacement. For points inside the ellipse and on the right boundary of the ellipse in the observation, this distance will be less, as it is always the distance to the left. The average of these distances over all the points in the ellipse of the forecast is $dist_{Ob}$. The final step is to average $dist_{OV}$ and $dist_{Ob}$. This is the verification distance for forecast1.

In this case, the verification metric penalizes the highly displaced forecast2 the most and the over forecast of forecast5 nearly as much, demonstrating the impact of $dist_{DV}$ and $dist_{OV}$ respectively. As would be expected, the forecast exhibiting a small displacement (forecast1) is declared the best and the value of $metr_V$ (46 pixels) is close to the known displacement of 50 pixels. The over forecast in forecast3 and rotation in forecast4 receive intermediate scores. It should be noted that $metr_V$ lies in the range of 0, and so there is no way to specify the threshold beyond which a forecast is bad. A forecast that exhibits a 50-pixel displacement may be considered bad for some applications, tolerable for others and very useful in some cases. Comparing these results with that of Keil and Craig (2009) and Davis et al. (2006), we suggest that our ranking (01, 04, 03, 05, 02) is more understandable than that of either Keil and Craig (2009) (01, 02, 05, 04, 03) or of Davis et al. (2006) (04, 03, 05, 02, 01). Note that our ranking places the slightly displaced figure in forecast1 highest whereas the method of Davis et al. (2006) favors the much larger over forecast in forecast4 because it happens to overlap slightly with the observation.

To calculate the comparison distance, we find the points whose values are equal to one and their corresponded coordinates in the matrix after applying the

threshold. For each point in the observation and forecast, their distance is calculated using the Euclidean formula (Equation 1.1). The average of all distances demonstrates the comparison distance between the observation and forecasts. From the results, it could be seen that the comparison method does not satisfy the positive property of a metric, because the distance from observation to itself is not zero. However, the comparison method provides an alternative way to compare different forecasts. The smaller the distance means the better the forecast.

1.5.2 Perturbed cases

The "perturbed" set of cases from the Inter-comparison Project (Ahijevych et al., 2009) consists of observed data from the 2005 NSSL/SPC Spring Experiment described in Kain et al. (2008). The observed data are subjected to various transformations such as shifting the entire image by a known number of pixels or multiplying the pixel value by a known amount. The observation and forecasts are shown in Figure 1.3. The results of verification are shown in Table 1.2.

We also analyze observed data and model that runs from the 2005 NSSL/SPC Spring Experiment described in Kain et al. (2008) and use for inter-comparisons by Gilleland et al. (2009). The observed data from May 14, 2005 are compared with 24 hour forecasts of 1 hour rainfall accumulation carried out on May 13, 2005.

The observations and model forecasts (from the CAPS, NCAR and NCEP models) are shown in Figures 1.4 and 1.5. The images cover the lower 48 states of the United States. The NCEP model forecast is produced at the National Centers for Environmental Prediction (NCEP) using a Weather Research and Forecasting (WRF)

model whose core is a Nonhydrostatic Mesoscale Model (Janjic et al., 2005) with a 4.5 km grid spacing and 35 vertical levels. The NCAR model forecast was produced at the National Center for Atmospheric Research using the Advanced Research WRF (ARW; Skamarock et al., 2005) core with a 4 km grid spacing and 35 vertical levels. The CAPS is produced at the Center for Analysis and Prediction of Storms at the University of Oklahoma (also using the ARW core) with a 2 km grid spacing and 51 vertical levels. All three forecast systems use initial and lateral boundary conditions from the North American Mesocale Model (Rogers et al., 2009). The observations are from the Stage II rainfall accumulation dataset produced by NCEP (Baldwin and Mitchell, 1998).

Since the verification metric depends on the threshold used to evaluate the image, we show the impact of filter the image by illustrating the images at two thresholds. The results of the verification are shown in Tables 1.4 and 1.5.

1.6 Discussion and Summary

It should be pointed out that the verification metric introduced in this chapter emphasizes the location error at the expense of fine structures in the forecast since the initial step, of converting the fields to binary by applying a threshold, treats all pixels above the threshold identically regardless of how much above the threshold the pixel's value is. It is possible to use a graph to derive the variation of the metric by threshold and compute a scalar metric such as the area under the curve to obtain a simple scalar

metric that takes into account all the pixel values. For simplicity of analysis, however, we concentrate on a single threshold in all the experiments.

In this chapter we suggest that by using metrics for spatial verification, it is possible to use a simple scalar number to capture the goodness of a forecast even if there is no pixel-to-pixel correspondence. Further, we devise a verification metric (Equation (1.9)) and show that it was suitable for verifying model forecasts. In the end, we give a comparison model to compare the distances from different forecast to the observation. This model is efficient when we need to choose the best forecast out from various forecasts.

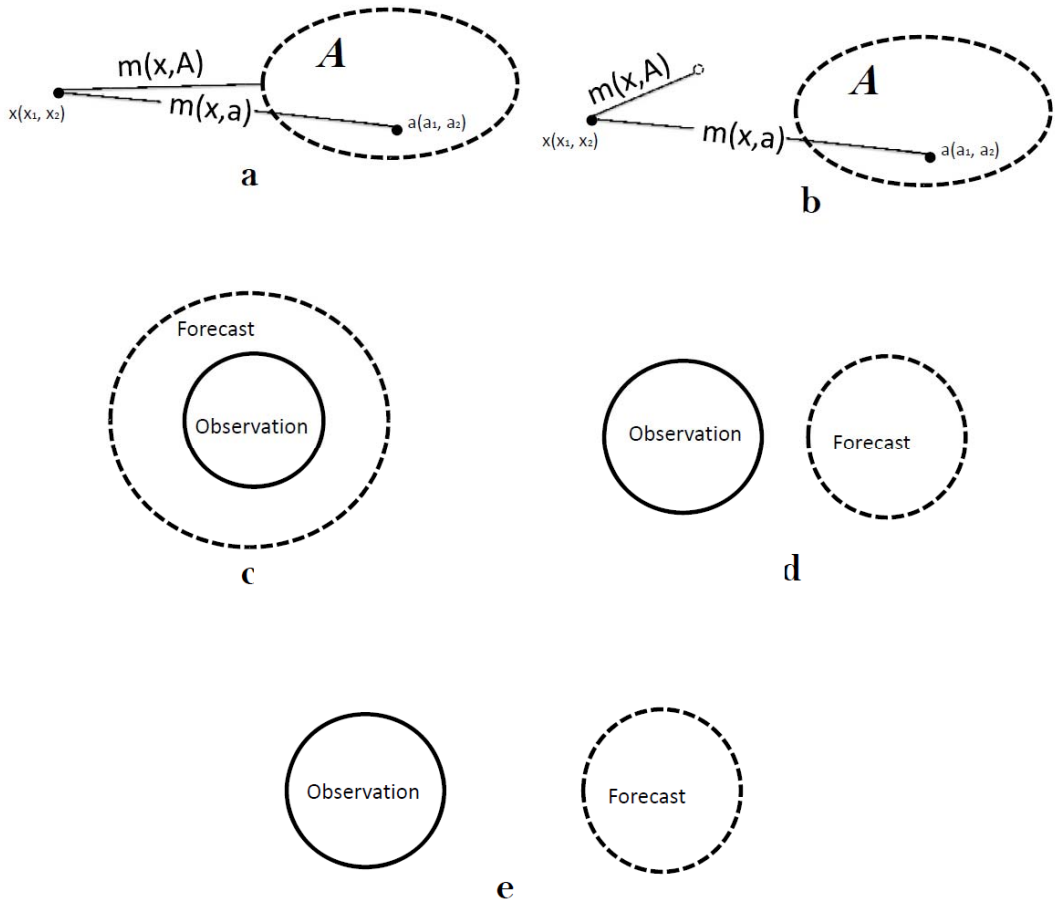


Figure 1.1 Schematic plots of observation and forecast fields of different scenarios.

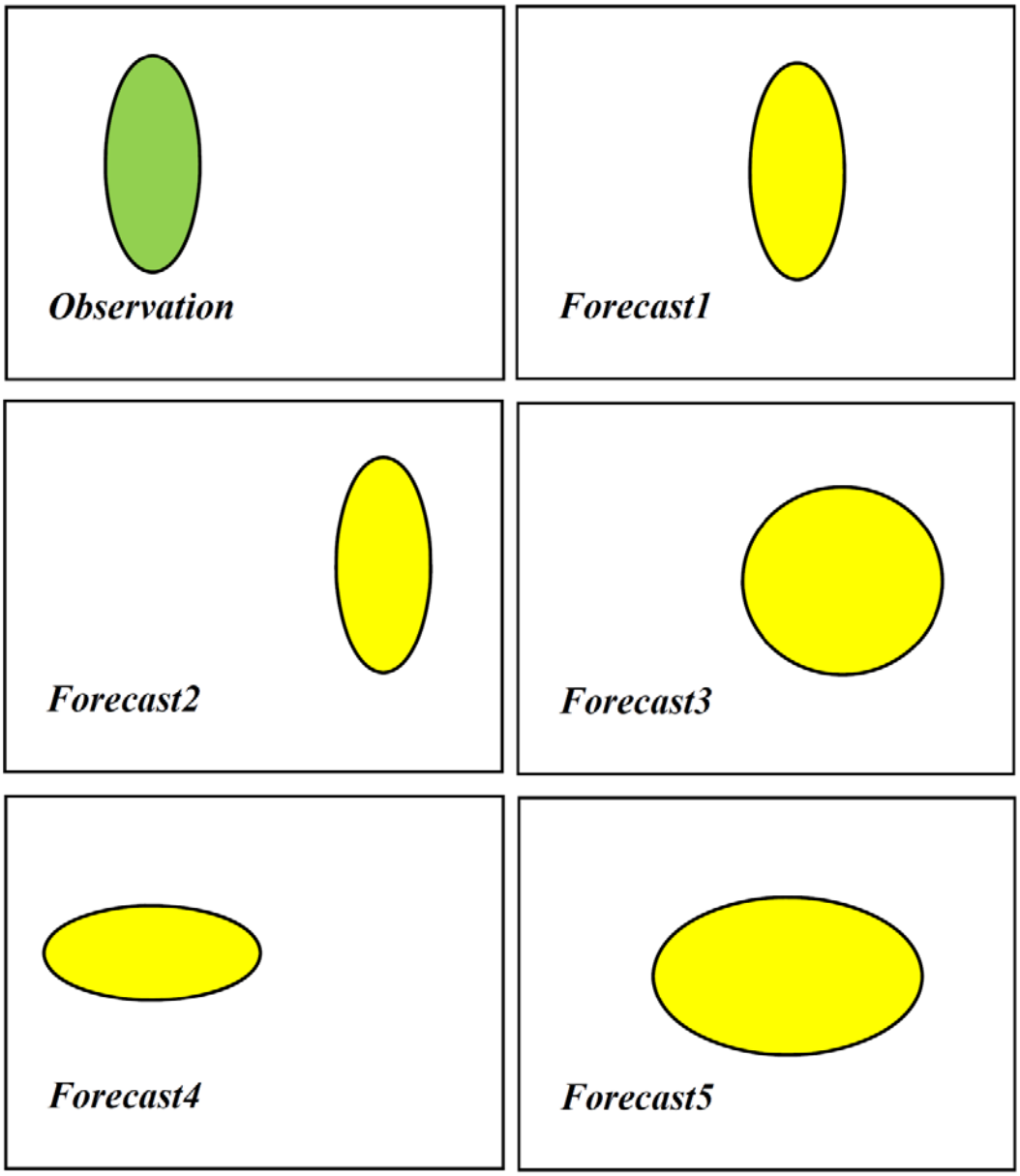


Figure 1.2 Verification metric for the geometric cases.

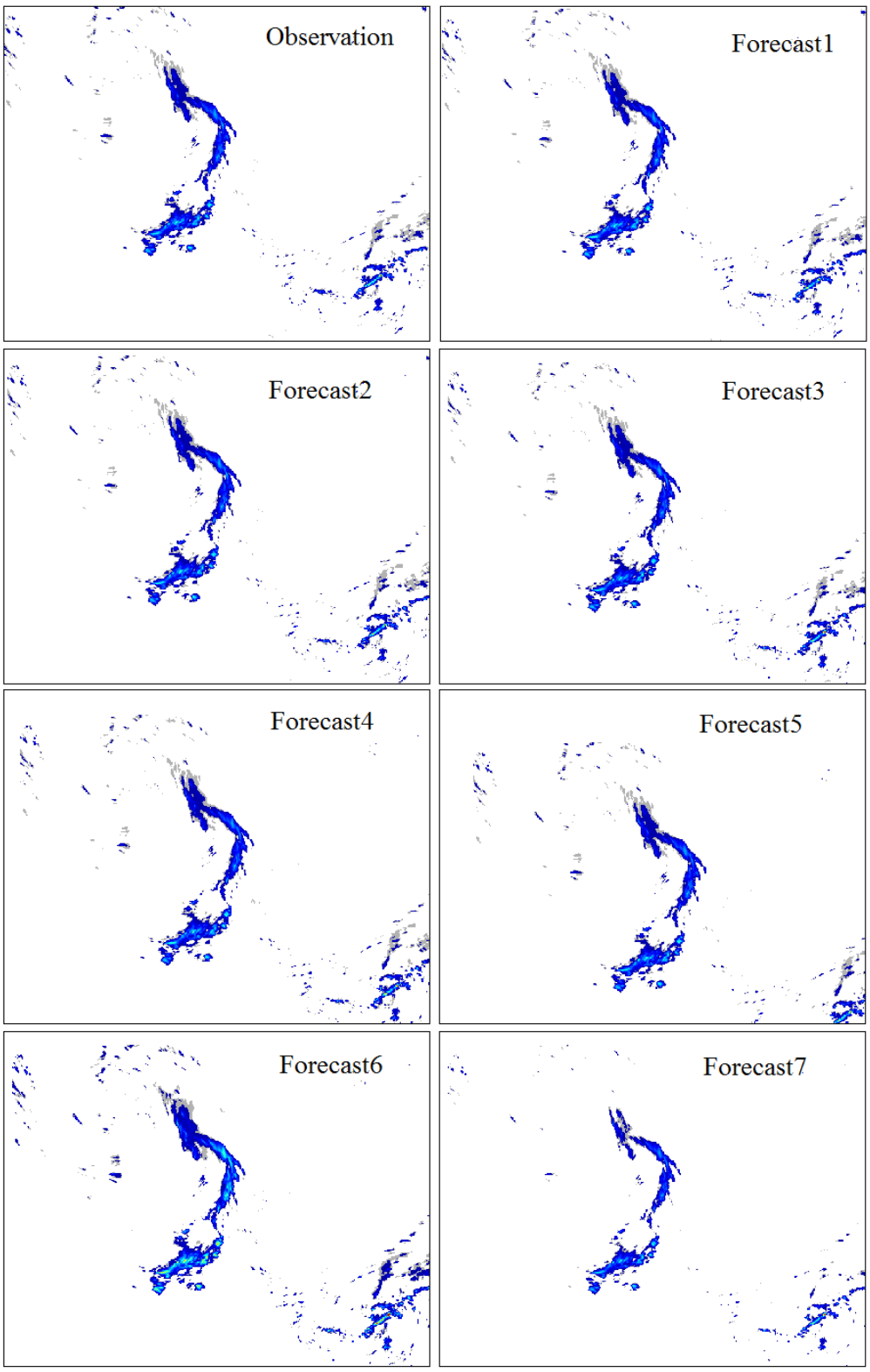


Figure 1.3 Verification metric for perturbed images from Ahijevych et al. (2009) with threshold at 0 mm.

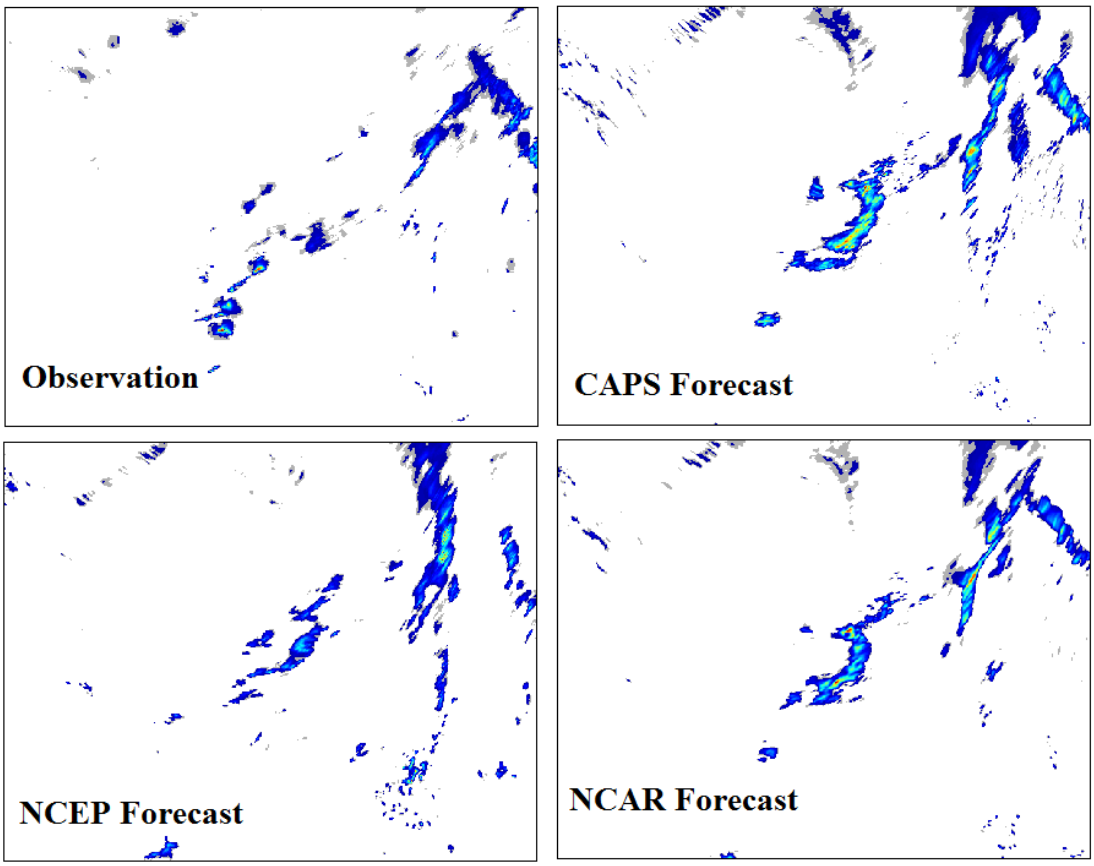


Figure 1.4 Verification metric for 24 hour precipitation forecasts valid for May 14, 2005 against the observations on that day with threshold at 0mm.

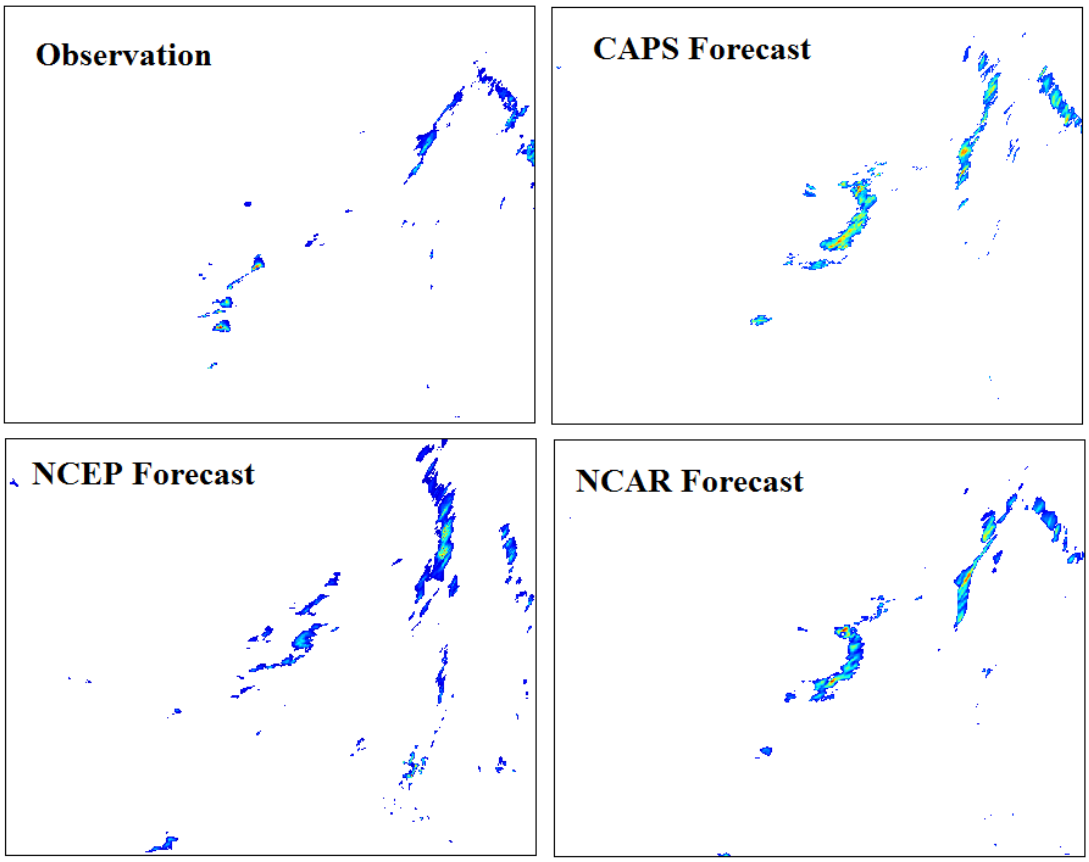


Figure 1.5 Verification metric for 24 hour precipitation forecasts valid for May 14, 2005 against the observations on that day with threshold at 20mm.

Data Set	$metr_V$	$\text{dist}_{\text{comp}}$
Observation	0	45
Forecast1	46	80
Forecast2	120	157
Forecast3	98	132
Forecast4	84	110
Forecast5	112	138

Table 1.1 Verification metric and Comparison distance of geometric images.

Data Set	$metr_V$,0 mm	$\text{dist}_{\text{comp}}$
Observation	0	78
Forecast1	80.0	162
Forecast2	91.8	180
Forecast3	103.6	185
Forecast4	116.6	192
Forecast5	131.7	231
Forecast6	103.6	165
Forecast7	103.3	172

Table 1.2 Verification metric and Comparison distance of perturbed images from Ahijevych et al. (2009) when considered at thresholds of 0 mm.

Data Set	$metr_V$, 20 mm	$\text{dist}_{\text{comp}}$
Observation	0	57
Forecast1	18.5	81
Forecast2	20.9	89
Forecast3	24.7	95
Forecast4	33.7	110
Forecast5	53.8	128
Forecast6	29.6	126
Forecast7	24.3	117

Table 1.3 Verification metric and Comparison distance of perturbed images from Ahijevych et al. (2009) when considered at thresholds of 20 mm.

Data Set	$metr_V$	$\text{dist}_{\text{comp}}$
Observation	0	128
CAPS Forecast	106.5	217
NCAR Forecast	105.0	226
NCEP Forecast	114.9	231

Table 1.4 Verification metric and Comparison distance of forecasts in May 15 2005 with threshold at 0mm.

Data Set	$metr_V$	$\text{dist}_{\text{comp}}$
Observation	0	112
CAPS Forecast	46.8	241
NCAR Forecast	47.5	238
NCEP Forecast	53.8	262

Table 1.5 Verification metric and Comparison distance of forecasts in May 15 2005 with threshold at 20mm.

CHAPTER 2
Forecasting Rotating Storms

In this chapter we introduce a systematical method for tracking both the translation and the rotation of tropical storms (such as hurricanes and typhoons) based on remotely sensed images. In our approach, we first obtain certain global properties of given images based on a new designed active contour method. From these global properties, we derive the rotation angle as well as the translation of the center for the movement of the object.

2.1 Introduction

In this chapter, we propose a new approach to determine the translation and rotation of tropical storms, such as hurricanes and typhoons.

Over short time periods, weather forecasts are based on statistical properties and over long time periods, they are based on numerical models. The change-over in skill occurs at 3-6 hours depending on the weather phenomenon: usually under 3 hours, statistical models outperform and over 6 hours, numerical models outperform. A key component in statistical weather prediction systems, known as "nowcasting" systems, is the ability to track a storm from remotely sensed images (radar or satellite) so as to study trends in various properties of the storm and to extrapolate the storm location in the future. Storm tracking, accordingly, has been studied extensively, see, e.g. Dixon et. al (1993), Johnson et. al (1998), Tuttle and Gall (1999) and Lakshmanan et. al (2003). A pattern matching approach is used to determine the velocity of a storm and to then

extrapolate the storm position. Corresponding patterns in successive images are identified and used to determine storm attributes in order to create a machine intelligence model of storm behavior (Lakshmanan et. al 2008).

Tropical storms such as hurricanes and cyclones rotate about their centers while also exhibiting a translation of position. For successful nowcasting of tropical storms, it is necessary to be able to capture the rotational aspects of this movement. Otherwise, rain bands on the periphery of the cyclone will correspond to the wrong location in the previous image causing statistical properties of the rain band to be wrong.

Figure 2.1 (a) shows hurricane Ike on Sep. 13, 2008 as it makes landfall near Houston around 02.30 UTC. An hour and a half later, the rain band has rotated about 20 degrees and is now directly over the city (Figure 2.1 (b)). The nowcasting problem is to capture both the slow motion towards the northwest of the entire hurricane and the fast rotation of the rain band on the shore. To date, nowcasting systems have been able to capture either one or the other but not both. Even methods such as Tuttle and Gall (1999) that aim to capture the winds of the hurricane are not useful for nowcasting at the timescale (1.5 hours) that is needed because the winds at the rain band section that would be estimated at 02.30 UTC would be eastward and a straight-forward linear extrapolation will put the rain band further east, not north. What is needed is to capture the rotation of the storm. Larger-scale methods tend to capture the movement of just the eye of the hurricane and capture the northwest movement, but not the smaller-scale rotation which is essential to a better nowcast of hurricanes. Nowcasting techniques so far have limited themselves to identifying the gross movement, ignoring the rotation. One of the major reasons is due to the technical difficulty in estimating the center of

location and angle of rotation from frames of remotely sensed images.

In this chapter we introduce a systematically novel approach to address the problem. Our mathematical approach is capable of tracking rotating objects efficiently. One of our key observations is that the rotation angle of a rigid motion does not depend on the choice of coordinate systems (which might be viewed as an intrinsic property of the storm). This enables us to design a novel mathematical approach to compute the angle. The location of the center shall follow easily from the angle of the rotation. In this chapter, we shall describe our approach in details. We apply our method to track synthetic rotating motions, as well as to track a real hurricane for radar images.

Our approach consists of two key steps. First of all we need to obtain various intrinsic characteristics of an image (may not necessary be the edges in the image) at a given time. Here we shall use active contour methods with a new energy functional. In the past decade, the active contour method has been widely used in image processing, in particular in medical image analysis, though it has not been widely used for weather forecasting. In our new energy functional we introduce certain parameters motivated by the structure of storms in radar images. These adjustable parameters play essential role in our design of computing angular motion. In the second step, we develop an algorithm which can automatically extract the rotating angle and center from those characteristics obtained in the first step.

2.2 Intrinsic characteristics of objects

Intrinsic characteristics of a storm, in term of a radar or satellite image, could include the boundary of the storm, the locations of maximal intensity, or other data

involving derivative or even higher order derivatives of the remotely sensed variables. To find the edge or other intrinsic characteristic of the image, we shall use the modern active contour method which based on certain global properties of the image. In M. Zhu and P. Zhang (2009), a new energy functional is introduced for radar image segmentation. The advantage of the new functional comparing with other functional for active contour, in particular for radar image with noise, was also discussed in M. Zhu and P. Zhang (2009). Here the same functional to capture the essential characteristics of a storm is used.

2.2.1 Review of active contour method

Active contour is the procedure that we use to deform a given curve so that a given functional of the curve will achieve its local minimal value. This method is widely used recently in computer vision in seeking the edges or contours of given images. See, for example, Mumford and Shah(1989), Kass, Witkin and Terzopoulos(1988) , Caselles, Kimmel and Sapiro(1997), and Chan and Vese(2001).

Let $u_0(x,y)\colon \Omega \to \mathbb{R}$ be the gray level function of a given image. If u_0 is smooth, then the edge of the image consists of those points (x,y), where $|\nabla u_0|$ is relatively large.

The geometric contour aiming to detect edge automatically is based on the size of $|\nabla u_0|$. Let $C(s)\colon [0.L] \to \mathbb{R}^2$ be a closed curve, where s is its arc length parameter. One can introduce an edge-detecting function $g\colon \mathbb{R} \to \mathbb{R}$ so that $g(z) \to 0$ as $z \to \infty$. A typical example of such function is given by

$$\mathrm{g(z)} = \frac{1}{1+\mathrm{z}^2}$$

(2.1)

One can define the energy functional of C by

$$I_1(C) \coloneqq \int_0^L g(|\nabla u_0(C(s))|)ds. \quad (2.2)$$

In order to find the edge of image u_0, one can compute the local minimal for I_1 (the geometric active contour model (1997)):

$$I_1(edge) = inf_C \ I_1(C) \quad (2.3)$$

There are other models in seeking the edge of a given image. For example, the snake model (V. Lakshmanan, 2003) is to introduce, for a parameterized curve $C(p)$: $[0,1] \to \mathbb{R}^2$, the following energy functional:

$$I_2(C) = \int_0^1 |C'(p)|^2 \, dp + \beta \int_0^1 |C''(p)| \, dp - \lambda \int_0^1 |\nabla u_0(C(p))|^2 \, dp. \quad (2.4)$$

where $\alpha,\ \beta,\ \lambda$ are all positive parameters. The first two terms represent the internal energy of the image, which usually are used to smooth the curve; The third term represents the external energy, serving as the indicator for edge. The edge of the image then can be found by minimizing I_2:

$$I_2(edge) = \inf_C I_2(C). \quad (2.5)$$

When the given image $u_0(x, y)$ is not smooth, the edge of the image is not well defined based on the derivative of the gray level function. Intuitively, one way to

determine the edge of a non smooth image is to identify the boundary of different groups. To identify such boundary, one can use Chan-Vese energy (Chan and Vese, 2001):

$$I_3(C, c_1, c_2) \coloneqq \int_{inside(C)} |u_0 - c_1|^2 \, dxdy + \int_{outside(C)} |u_0 - c_2|^2 \, dxdy + \mu \cdot \big(length(C)\big) + \nu \cdot \Big(Area\big(inside(C)\big)\Big). \quad (2.6)$$

where c_1, c_2 are constants to be adjusted in iteration, μ and ν are fixed parameter. The last two terms are smoothing terms. The edge is again sought by minimizing $I_3(C, c_1, c_2)$:

$$I_3(edge, c_{1.*}, c_{2.*}) = \inf_{C, c_1, c_2} I_3(C, c_1, c_2). \quad (2.7)$$

2.2.2 Numerical implementation via level set method

To automatically detect the edge via an iteration scheme, one can introduce a family of curves $C(p,t)$: $[0,1] \times [0,\infty) \to \mathbb{R}^2$ and the deformation path. Numerically, such iteration can be realized via the powerful level set method of Osher and Sethian(1988). For example, level set method can be used for Chan-Vese model as follows. We introduce the Heaviside function and its derivative

$$\mathrm{H}(\mathrm{z}) = \begin{cases} 1 & if \quad z \geq 0, \\ 0 & if \quad z < 0, \end{cases} \qquad \delta(z) = \frac{d}{dz} H(z). \quad (2.8)$$

Embedding $C(p,t)$ as a nodal line of a smooth function $\Phi(x,y,t)$: $C = \{(x,y,t): \Phi(x,y,t) = 0\}$, we then can re-write the energy functional $I_3(C, c_1, c_2)$

as

$$J_3(\Phi, c_1, c_2)$$

$$= \int_\Omega |u_0 - c_1|^2 H\big(\Phi(x,y)\big) dxdy + \int_\Omega |u_0 - c_2|^2 (1 - H\big(\Phi(x,y)\big))\, dxdy$$

$$+ \mu \int_\Omega \delta\big(\Phi(x,y)\big) |\nabla\Phi(x,y)|\, dxdy$$

$$+ \nu \int_\Omega H\big(\Phi(x,y)\big)\, dxdy. \qquad (2.9)$$

For fixed Φ, minimizing $J_3(\Phi, c_1, c_2)$ with respect to c_i yields:

$$\begin{cases} c_1(\Phi) = \text{average}(u_0) \ \text{ in } \ \{\Phi \geq 0\} \\ c_2(\Phi) = \text{average}(u_0) \ \text{ in } \ \{\Phi < 0\} \end{cases} \qquad (2.10)$$

Once c_1 and c_2 are fixed, we minimize J_3 via deforming Φ along the gradient direction of energy functional:

$$\begin{cases} \dfrac{\partial\Phi}{\partial t} = \delta(\Phi)\{\mu \cdot div\left(\dfrac{\nabla\Phi}{|\nabla\Phi|}\right) - \nu - (u_0 - c_1(\Phi))^2 + (u_0 - c_2(\Phi))^2\} \ \text{ in } \ \Omega \times (0,\infty) \\ \Phi(x,y,0) = \Phi_0(x,y) \quad \text{in} \quad \Omega \\ \dfrac{\delta(\Phi)}{|\nabla\Phi|}\dfrac{\partial\Phi}{\partial n} = 0 \ \ on \ \ \partial\Omega \times (0,\infty) \end{cases} \qquad (2.11)$$

where the initial data $\Phi_0(x,y)$ is chosen as a signed distance function to a given initial curve $C(p,0)$.

In Zhu and Zhang (2009), a new model for radar image segmentation was introduced in order to remove meteorological noise (radar noise, insect noise, etc.). That model is based on the observation that biological echoes usually have relatively low intensity to severe storms, and radar signals for storm are usually uniform in certain region. They consider the following modified Chan-Vese functional:

$$I_4(C,c) \coloneqq \int_{inside(C)} |u_0 - \alpha \cdot M|^2 \, dxdy + \int_{outside(C)} |u_0 - c|^2 \, dxdy$$

$$+ \quad \mu \cdot \big(length(C)\big) + \nu \cdot \Big(Area\big(inside(C)\big)\Big) \tag{2.12}$$

where $M = max_{x\in\Omega} u_0(x)$, c is a constant to be adjusted in iteration, α, μ and ν are adjustable parameters. For a reasonable chosen parameter α, it is shown in Zhu and Zhang (2009) that Model (2.12) is more robust than Chan and Vese's model for radar image segmentation. In next section we will show another important role parameter α will play in computing the rotation angle of a rigid motion.

Again, we can implement above model via level set method as follows. Embedding $C(p)$ as a nodal line of a smooth function $\Phi(x,y)$: $C = \{(x,y): \Phi(x,y) = 0\}$such that $\Phi(x,y) > 0$ insider the curve and $\Phi(x,y) < 0$ outsider the curve, we can re-write the energy functional $\mathrm{I}_4(C,c)$ as

$$\mathrm{J}_4(C,c) = \int_\Omega |u_0 - \alpha \cdot M|^2 \, H\big(\Phi(x,y)\big)dxdy$$

$$+ \int_\Omega |u_0 - c|^2 (1 - H\big(\Phi(x,y)\big)) \, dxdy$$

$$+ \mu \int_\Omega \delta\big(\Phi(\mathrm{x},\mathrm{y})\big)|\nabla\Phi(\mathrm{x},\mathrm{y})| \, \mathrm{dxdy}$$

$$+ \nu \int_\Omega \mathrm{H}\big(\Phi(\mathrm{x},\mathrm{y})\big) \, \mathrm{dxdy}. \tag{2.13}$$

where $H(\cdot)$ is the Heaviside function defined in Section 2.2.1. For fixed Φ, minimizing $\mathrm{J}_4(\Phi,c)$ with respect to c yields

$$c(\Phi) = \mathrm{average}(\mathrm{u}_0) \text{ in } \{\Phi < 0\}.$$

(2.14)

To derive the first variation of the functional, we consider slightly regularized version of functions H_ε and $H'_\varepsilon = \delta_\varepsilon$ such that$H_\varepsilon \in C^\infty(\overline{\Omega})$, $H_\varepsilon \longrightarrow H$ and $\delta_\varepsilon \longrightarrow \delta$, and the modified functional:

$$\begin{aligned} \mathrm{J}_{4,\varepsilon}(\Phi, c) = & \int_\Omega |u_0 - \alpha \cdot M|^2 H_\varepsilon\big(\Phi(x,y)\big) dxdy \\ & + \int_\Omega |u_0 - c(\Phi)|^2 (1 - H_\varepsilon\big(\Phi(x,y)\big))\, dxdy \\ & + \mu \int_\Omega \delta_\varepsilon\big(\Phi(\mathrm{x},\mathrm{y})\big) |\nabla\Phi(\mathrm{x},\mathrm{y})|\, \mathrm{dxdy} \\ & + \nu \int_\Omega H_\varepsilon\big(\Phi(\mathrm{x},\mathrm{y})\big)\, \mathrm{dxdy}. \qquad (2.15) \end{aligned}$$

We can compute its first variation:

$$\begin{aligned} & < \delta J_{4,\varepsilon}, \psi > \\ & = \int_\Omega (u_0 - \alpha \cdot M)^2 \delta_\varepsilon\big(\Phi(x,y)\big)\psi dxdy - \int_\Omega (u_0 - c(\Phi))^2 \delta_\varepsilon\big(\Phi(x,y)\big)\psi\, dxdy \\ & + \mu \int_\Omega \delta_\varepsilon\big(\Phi(\mathrm{x},\mathrm{y})\big) \frac{\nabla\Phi}{|\nabla\Phi(\mathrm{x},\mathrm{y})|} \nabla\psi + \delta'_\varepsilon\big(\Phi(\mathrm{x},\mathrm{y})\big) |\nabla\Phi(\mathrm{x},\mathrm{y})| \psi \mathrm{dxdy} \\ & + \nu \int_\Omega \delta_\varepsilon\big(\Phi(\mathrm{x},\mathrm{y})\big)\psi\, \mathrm{dxdy}. \qquad (2.16) \end{aligned}$$

Thus we can minimize $J_{4,\varepsilon}$ via deforming Φ along the gradient direction of $J_{4,\varepsilon}$:

$$\begin{cases} \dfrac{\partial\Phi}{\partial t} = \delta_\varepsilon(\Phi)\{\mu \cdot \mathrm{div}\left(\dfrac{\nabla\Phi}{|\nabla\Phi|}\right) - \nu - (\mathrm{u}_0 - \alpha \cdot M)^2 + (\mathrm{u}_0 - \mathrm{c}(\Phi))^2\} \text{ in } (0,\infty) \times \Omega, \\ \Phi(x,y,0) = \Phi_0(x,y) \quad in \quad \Omega \\ \dfrac{\delta_\varepsilon(\Phi)}{|\nabla\Phi|} \dfrac{\partial\Phi}{\partial \mathrm{n}} = 0 \qquad \text{on} \quad \partial\Omega \end{cases} \qquad (2.17)$$

In practice, one can choose

$$H_\varepsilon(x) = \frac{1}{2}\left(1 + \frac{2}{\pi}\cdot \arctan\left(\frac{x}{\epsilon}\right)\right), thus \quad \delta_\varepsilon(x)$$

$$= \frac{1}{\pi}\cdot\frac{\varepsilon}{\epsilon^2 + x^2}. \qquad (2.18)$$

We use the above deformation on a synthetic image. Figure 2.2 (a) and (b) are two original images; Figure 2.3(a), (b) are the characteristics obtained via (2.17) for $\alpha = 0.5$. After we adjust $\alpha = 1$, we obtain other two characteristics Figure 2.3(c) and (d) for the Figure 2.2(a), (b), respectively. More details are given in Experiment 1 below. We shall address in next section the reason that we need to obtain two different characteristics for the same image. We need to point out that when we implement the algorithm, the standard re-initialization procedure is also used, see, e.g. Chan and Vese(2001).

2.3 Rigid Motion Tracking

Let $I(x): \Omega \subset \mathbb{R}^2 \longrightarrow \mathbb{R}_+$ be the initial image of an object we are tracking. After a fixed time, the object moves and the image is represented by its gray level function $I^1(x)$ in the same coordinate system. We assume the motion of the object is rigid. Then, if we write $I^1(x) = T(I(x))$, the transformation T consists of only a translation and a rotation with respect to a fixed point in the plane. Namely, for any x $\in \mathbb{R}^2$,

$$Tx = A(x - c) + d = A(x - s),$$

(2.19)

where

$$A(\theta) = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix}, s = c - A^{-1}d$$

$$:= \begin{pmatrix} s_1 \\ s_2 \end{pmatrix}. \qquad (2.20)$$

It is clear that the rotation angle θ does not depend on the center of the rotation, and the translation. In other words, we can always describe a rigid motion by a rotation (with respect to any point) with a fixed unique angle and a translation (the translation certainly depends on the choice of rotating center). See Figure 2.4 (a), a simple rotation; (b) a rotation plus a translation. Therefore, in order to track a rigid motion of an object, we shall first find the rotation angle, and then compute the translation of the motion after we choose the center of the rotation. Theoretically, the rotating angle can be computed based on the following simplified two-point rotating model.

Assume that $\mathrm{P}_1(\mathrm{x}_1, \mathrm{y}_1), \mathrm{P}_2(\mathrm{x}_2, \mathrm{y}_2)$, are two points in the first image $I(x)$. After a rigid motion, these two points move to point $\mathrm{Q}_1(\mathrm{u}_1, \mathrm{v}_1)$ and point $\mathrm{Q}_2(\mathrm{u}_2, \mathrm{v}_2)$ respectively. That is $\mathrm{Q}_1 = \mathrm{T}(\mathrm{P}_1), \mathrm{Q}_2 = \mathrm{T}(\mathrm{P}_2)$ are the two corresponding points in the second image $I^1(x)$. Then the rotation angle θ_* from vector $\mathrm{X} = \overline{\mathrm{P}_1\mathrm{P}_2}$ to vector $\mathrm{Y} = \overline{\mathrm{Q}_1\mathrm{Q}_2}$ can be computed as

$$\theta_* = cos^{-1}\frac{\mathrm{X}\cdot\mathrm{Y}}{|\mathrm{X}|\cdot|\mathrm{Y}|}. \qquad (2.21)$$

Here, we certainly assume that both vectors X, Y are not trivial. Thus the rotation operator $A_* := A(\theta_*)$ is known. If we further assume that the motion of the object is only a rotation around a center S, then we can compute the rotating center by

$$\mathrm{S} = (A_* - id)^{-1}\cdot(A_*\mathrm{P}_1 - \mathrm{Q}_1), \quad or\ \mathrm{S} = (A_* - id)^{-1}\cdot(A_*\mathrm{P}_2 - \mathrm{Q}_2) \qquad (2.22)$$

If we view the motion is a rotation with respect to a given center $\mathbf{c} \in \mathbb{R}^2$, then the translation (or called the displacement of the center) $d \in \mathbb{R}$ can be computed as

$$d = \mathrm{Q}_1 - A_*(\mathrm{P}_1 - \mathrm{c}), \ or \ d = \mathrm{Q}_2 - A_*(\mathrm{P}_2 - \mathrm{c}).$$

(2.23)

In practice, it is very difficult to identify four points $\mathrm{P}_1, \mathrm{P}_2, \mathrm{Q}_1, \mathrm{Q}_2$ in two real images such that $\mathrm{Q_i} = \mathrm{TP_i}$ for $i = 1,2$. In order to do so, we need to introduce certain particular characteristics for an image.

Definition (Center of mass): *Let* $f(x, y)$: $\Omega = (\mathrm{a}, \mathrm{b}) \times (\mathrm{c}, \mathrm{d}) \longrightarrow \mathbb{R}_+$ *be a gray level function for a given image. Its center of mass*(x_c, y_c) *is defined by*

$$x_c = \frac{\int_\Omega \ xf(x,y)dxdy}{\int_\Omega \ dxdy}, \quad y_c = \frac{\int_\Omega \ yf(x,y)dxdy}{\int_\Omega \ dxdy} \tag{2.24}$$

Definition (2.24) does not work very well for a generic image. For instance, it is sensitive to noise perturbation, even to the size of the domain. In applications, we can first simplify the image so that the gray image is a step function, and then compute the center. The simplification of an image can be realized via the active contour method which we discuss in previous section. See, e.g. Figure 2.3 (a) and (b). For convenience, the center of mass for the simplified image will be called an induced center of mass of the original image.

To use our two-point rotating model to compute the rotating angle, we need to find another induced center of mass for the given image. This can be again realized via our active contour method by choosing a different parameter α as being showed in Figure 2.3 (c) and (d).

Once we have two intrinsic induced centers of mass for each image, say,

P_1, P_2 for the first one, Q_1, Q_2 for the second one. We then have two vectors: $X = \overline{P_1P_2}$ and $Y = \overline{Q_1Q_2}$. Thus we can compute the rotation angle by (2.21), and the rotation center by (2.22). Here we see another advantage of using simplified image to compute the angle: even the background of the simplified images is changed, thus the denominators in (2.21) will change proportionally, but not the numerators, hence we obtain the same angle for the rotation.

2.4 Experiments

We first try a synthetic still image. We choose $\mu = 1$ and $v = 0$ in experiment 1.

Experiment 1. Figure 2.2(a) is an image whose gray level function is given by:

$$f(x,y) = \begin{cases} 1 & if \ \ (x-100)^2 + (y-40)^2 < 4 \\ \frac{1}{2} & if \ \ (x-100)^2 + (y-60)^2 < 4 \\ 0 & otherwise \end{cases}$$

(2.25)

The gray level function for Figure 2.2(b) is given by:

$$g(x,y) = \begin{cases} 1 & if \ \ (x-60)^2 + (y-100)^2 < 4 \\ \frac{1}{2} & if \ \ (x-80)^2 + (y-100)^2 < 4 \\ 0 & otherwise \end{cases}$$

(2.26)

First of all, using (2.17) for $\alpha = 0.5$ we obtain the first characteristics of images: Figure 2.3(a) and Figure 2.3(b). Using (2.17) for $\alpha = 1$ we obtain the second characteristics of images: Figure 2.3(c) and Figure 2.3(d). Applying (2.24) to Figure 2.3(a) and (b), we obtain the first induced centers for Figure 2.2(a) and (b): $P_1 = (100.5,50.5)$, $P_2 = (70.0,100.5)$, Applying (2.24) to Figure 2.3(c) and (d), we obtain

the second induced centers for Figure 2.2(a) and (b): $Q_1 = (100.5,40.5)$, $Q_2 = (60.5,40.5)$. Using (2.21) we estimate the rotating angle is 90° degree.

If we view the motion as one simple rotation (as indicated in Figure 2.4(a)), we can find its rotation center by (2.22): $S = (110.5,90.5)$.

If we view the motion as one rotation with respect to a given point $(100,50)$ and a translation (as indicated in Figure 2.4(b)), we can find the translation: $d = (-30,50)$.

Next, we apply our method to track rotating motions.

Experiment 2. We apply our method to track a synthetic rotating motion, see Figure 2.5(a). This is a rigid motion of an object, which rotates θ degree and then shifts the center along a given curve (in this case, the curve is a part of circle) with fixed rate m units in $x-$ direction. Ten images are put together in Figure 2.5 (b). The solid line in Figure 2.6 shows the simulation centers and the motion centers we capture using our algorithm. Figure 2.7 shows the forecast centers of rotation after we extrapolate from the motion centers and the rotating angles. There are two methods to using extrapolation: we can link previous two simulation centers and applying a linear extrapolation, then applying a rotation using the angle we obtained. This method yields forecast1 centers. Another method is: we link the motion center and the current simulation center, then rotate the line segment with respect to the motion center by the obtained rotating angle and to obtain forecast2 center. The motion centers as well as the rotating angles are given in Table 2.1.

Experiment 3. Our last experiment is to track a real hurricane: hurricane Bill 2009. The satellite images of Bill from 02AM, August 17 to 08AM, August 19 are given in Figure 2.8. The characteristics of these images are given in Figure 2.9 and Figure 2.10. The trajectory of the storm is labeled by green line and our computation yields the forecast path (we use linear extrapolation plus the rotating angle for nowcasting) labeled by blue boxes in Figure 2.11. The motion centers as well as the rotating angles from UT 1702 to UT 2002 are given in Table 2.2. The result indicates that the satellite images can be used to roughly track the path of the hurricane, as well as the rotating angles.

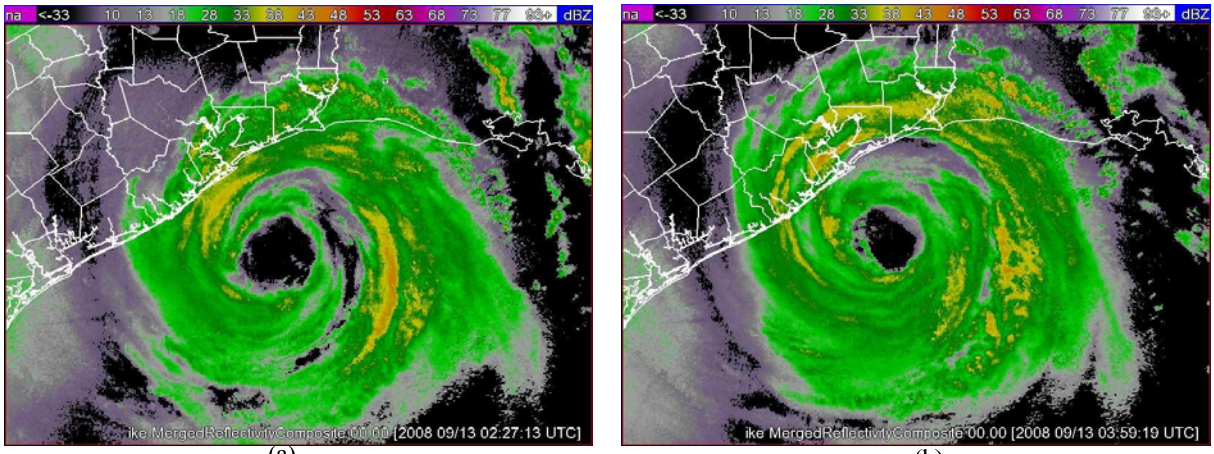
2.5 Conclusions

In this chapter, we propose a global approach to extract certain essential characteristics of given images based on active contour method. The obtained characteristics are used to classify a rigid motion, in particular, a rigid motion with rotation. Discussion on other motions will be addressed in future work.

There are certain limitations for our approach, even though they do not hurt its novelty. For instance, the choice of α and other parameters in (2.17) sometimes are essential in obtaining the characteristics of images. In fact, it is not too difficult to understand that we cannot obtain Figure 2.3(a) and (b) if α is chosen to be larger than 0.5 (for example, $\alpha = 0.8$). On the other hand, the experiments show that we cannot obtain Figure 2.3(a) and (b) even we choose $\alpha = 0.5$ but with large μ (for example, $\mu = 2$). The choice of these parameters for real problems may rely on certain experiences. In fact, one can certainly modify our energy functional further for proper

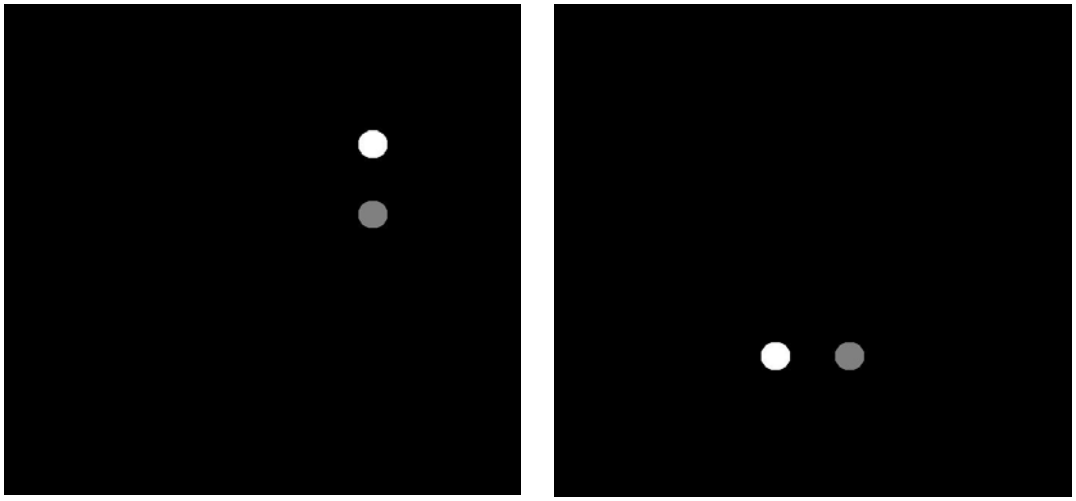
applications. Further, it is certainly possible to find better methods to catch the essential characteristics (which may not be the edges) of an image. This naturally becomes another challenging question in our further study.

Besides interesting applications of our method to nowcasting system of tropical storms, there are also many mathematical questions needed to be addressed in the future. For example, the global existence of flow equation, the uniqueness of the limit, the convergent speed of the flow, and the stability of the flow, are all interesting and nontrivial questions.



(a) (b)

Figure 2.1 Nowcasting the translation and rotation. (a) Initial landing of hurricane Ike, Sept. 13, 2008; (b) One and half hours later image.



(a) (b)

Figure 2.2 Synthetic images. (a) Original image; (b) Image after motion

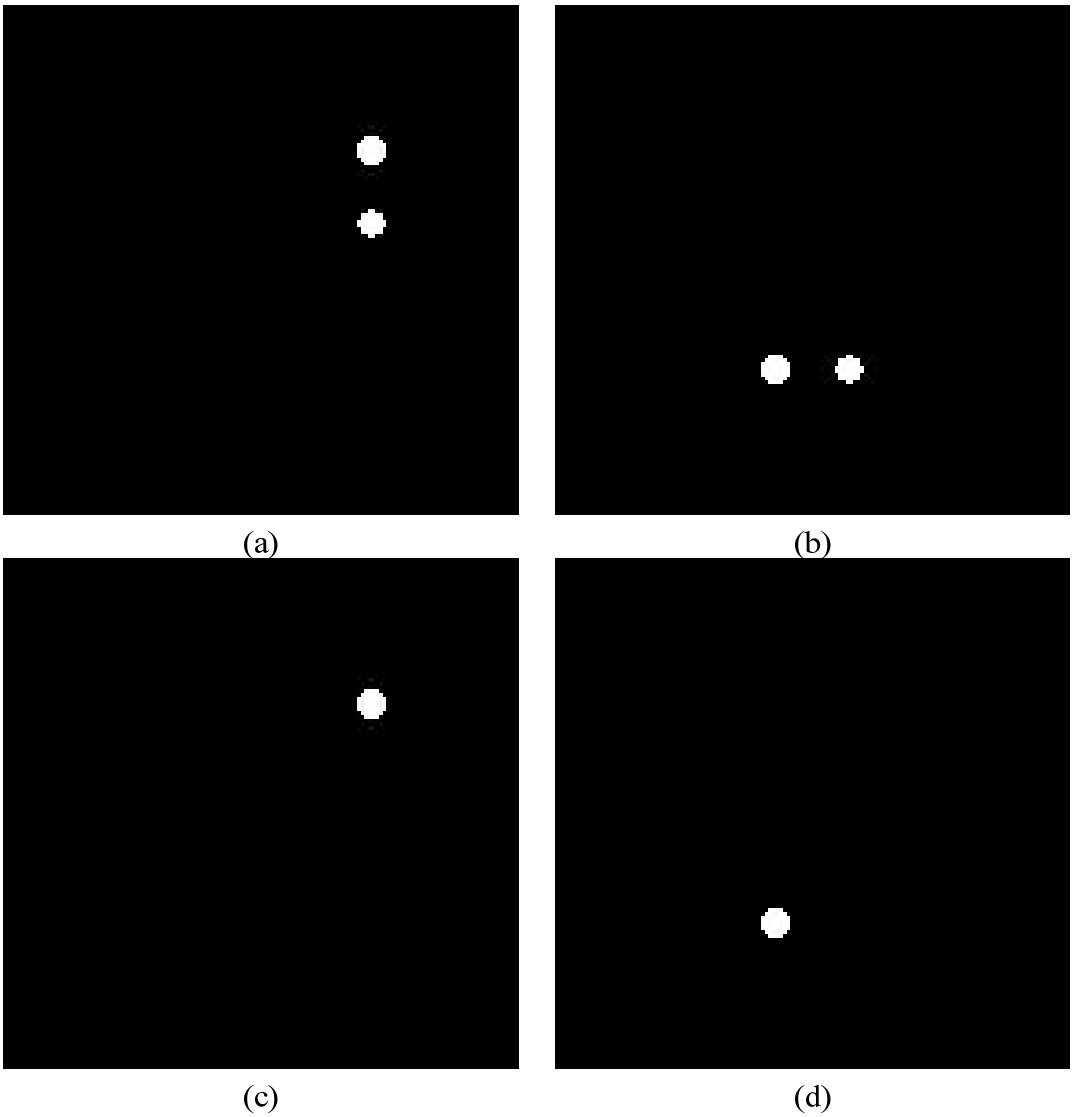
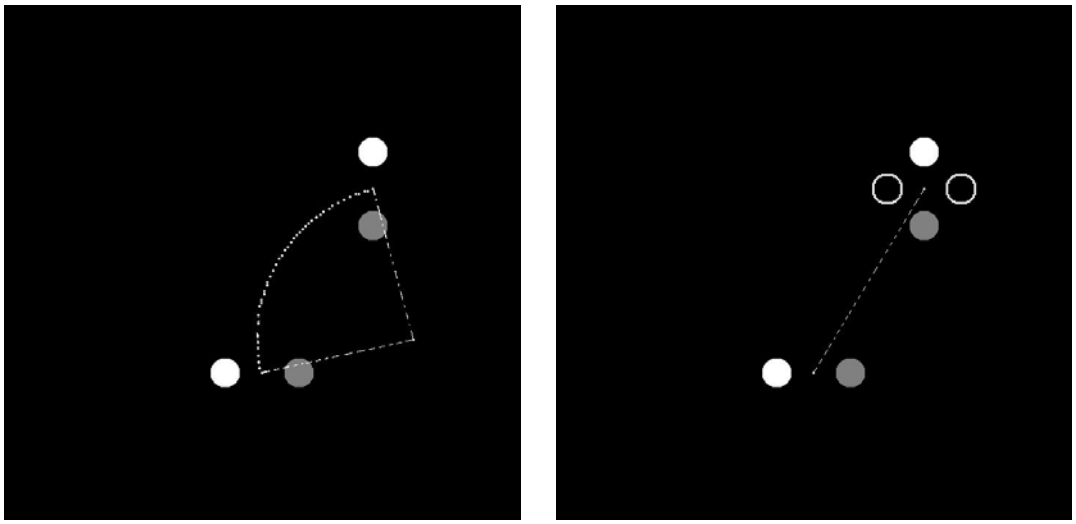
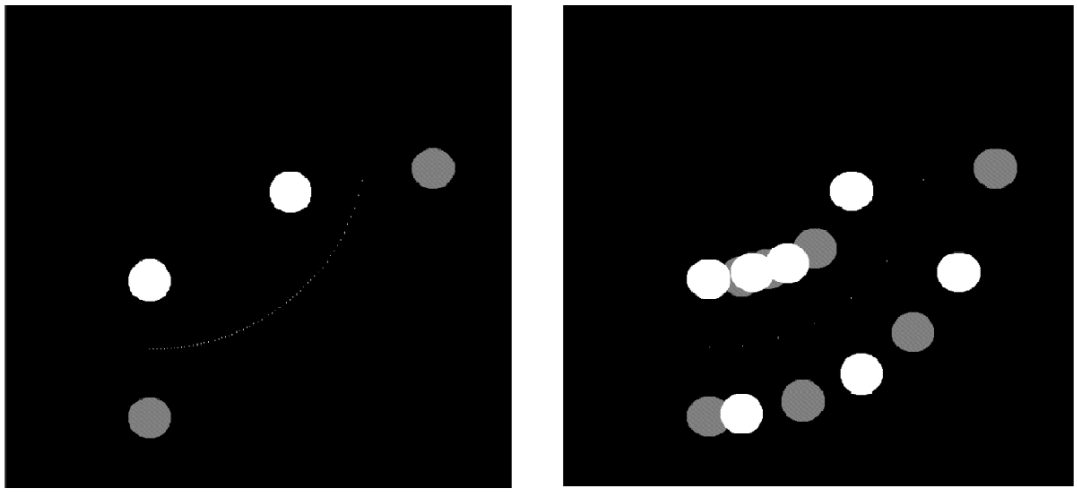


Figure 2.3 Two characteristics. (a) First characteristic for Figure 3.2(a); (b) First characteristic for Figure 3.2(b); (c) Second characteristic for Figure 3.2(a), (d) Second characteristic for Figure 3.2(b).



(a) (b)

Figure 2.4 Two descriptions of one rigid motions. (a) A simple rotation; (b) A translation after rotation.



(a) (b)

Figure 2.5 A synthetic rotating motion. (a) Synthetic rotating motion: the first moment and the last moment; (b) Ten moments of the motion in one chart.

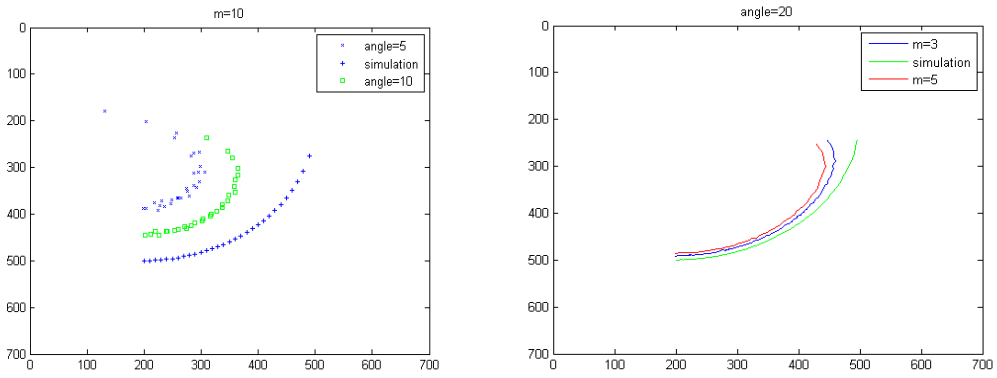


Figure 2.6 Motion centers VS rotating centers.

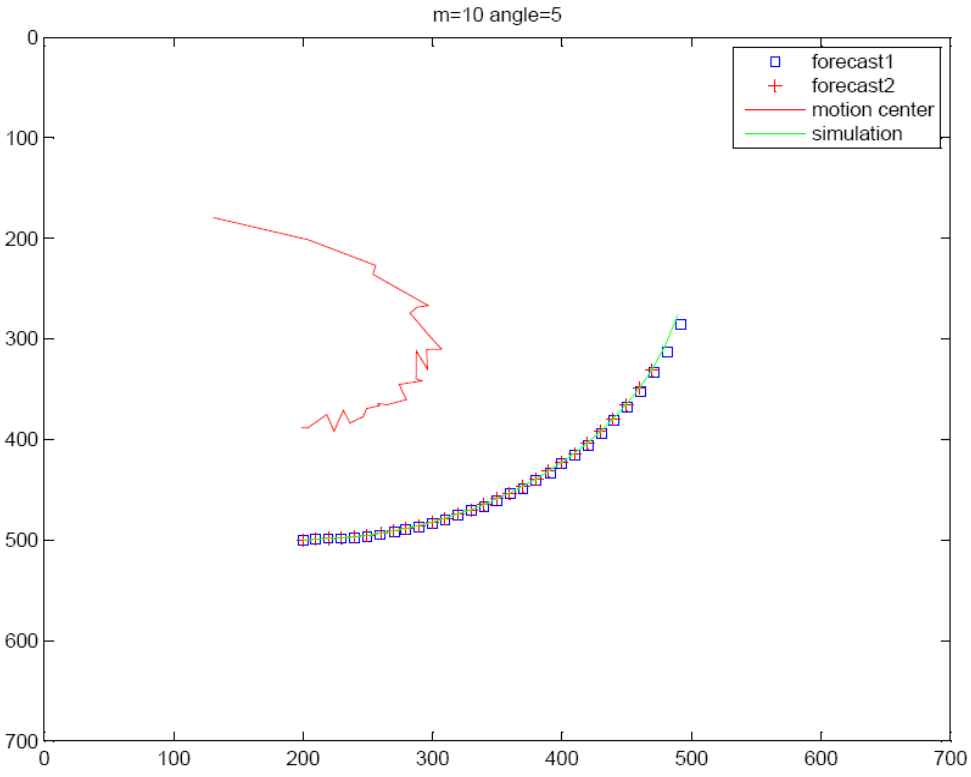


Figure 2.7 Forecasting via extrapolation.

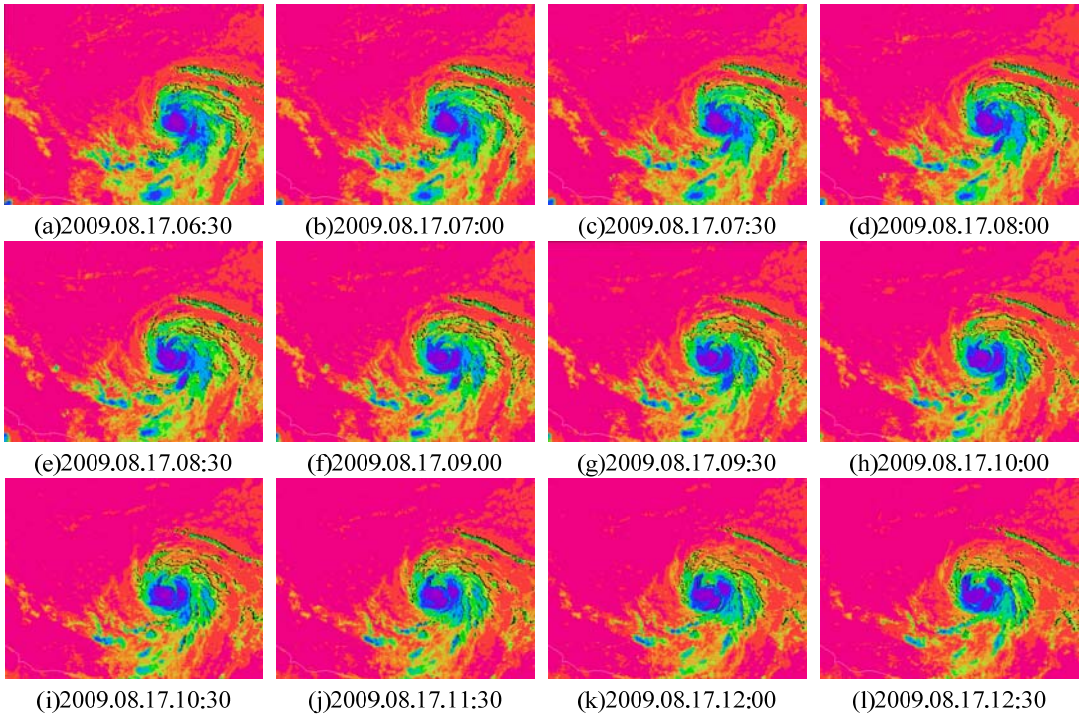


Figure 2.8 Hurricane Bill approaches Florida coast, August 17-19, 2009

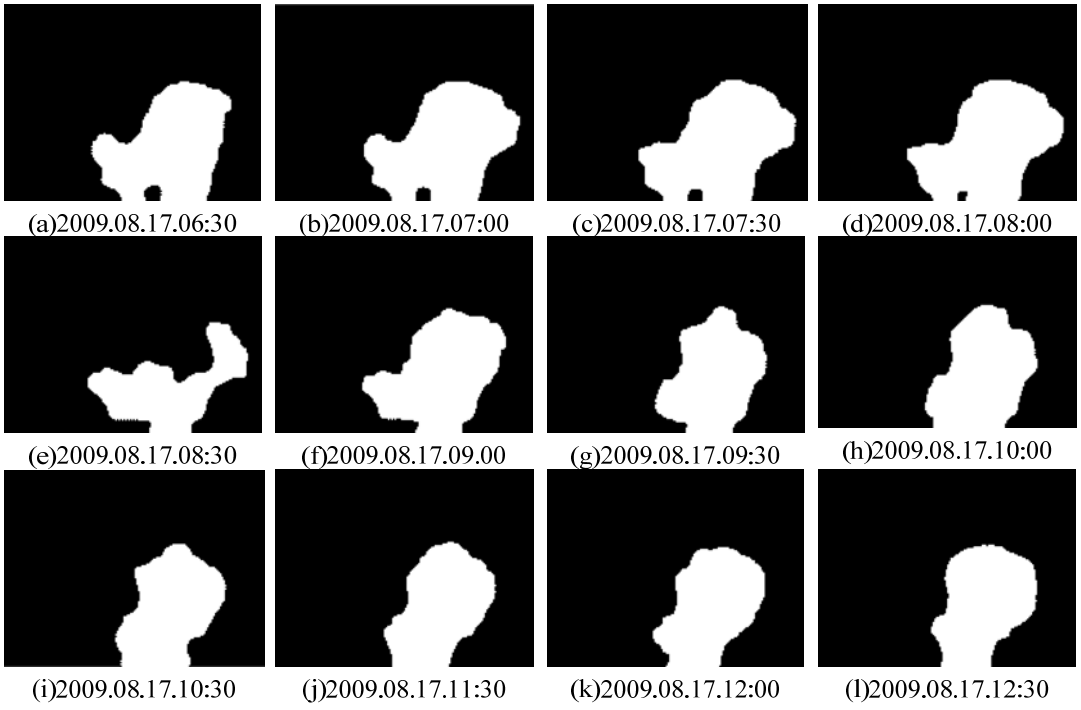


Figure 2.9 First characteristics of images, where $\alpha = 0.9$

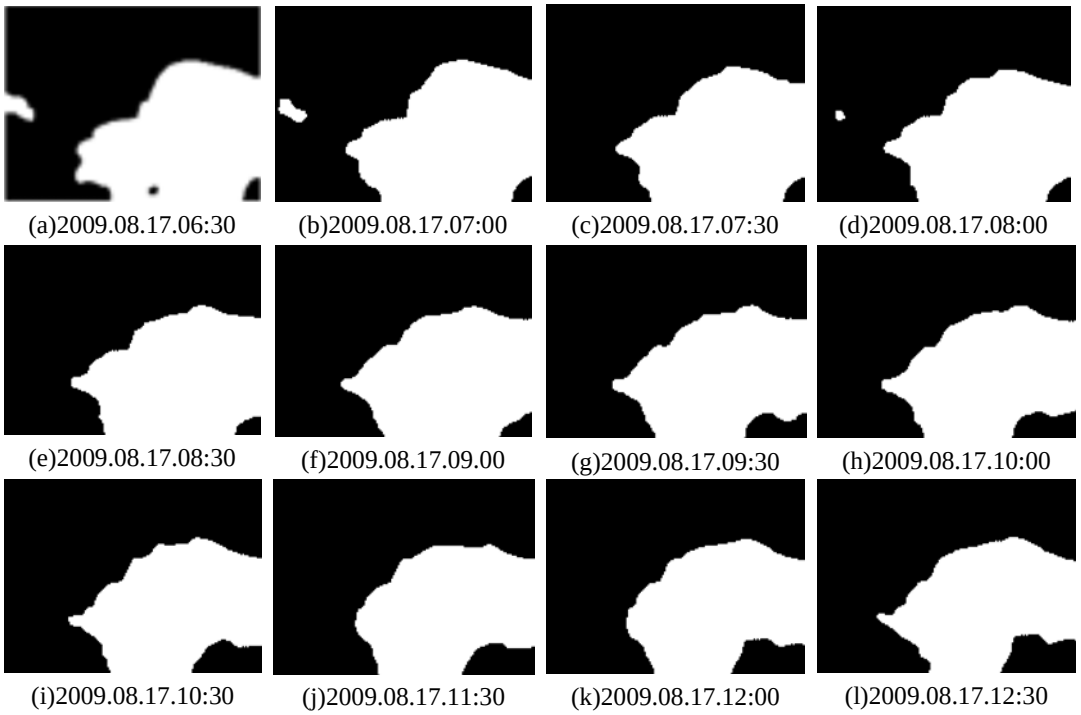


Figure 2.10 Second characteristics of images, where $\alpha = 0.7$

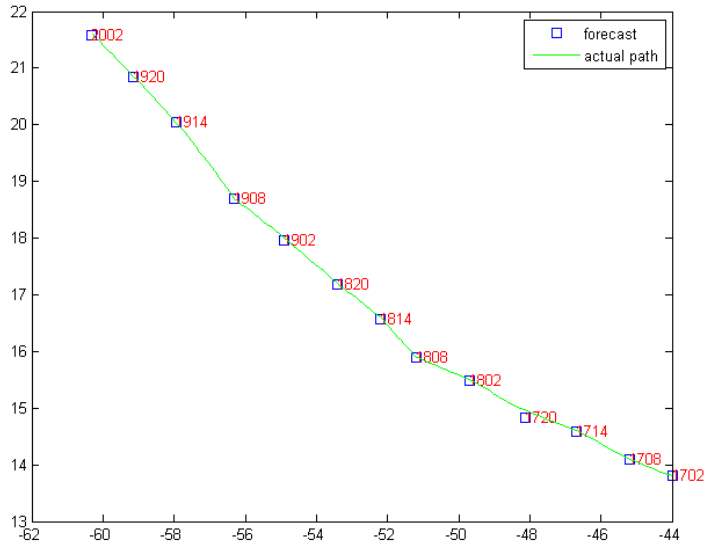


Figure 2.11 Tracking Storm Bill. The observed path comes from http://www.nhc.noaa.gov/2009atlan.shtml. Storm eyes VS centers of motion. The two nowcasting centers follow from two formulas in (2.22).

Position	Motion center1	Motion center	Angle
1	(198.8864,388.3416)	(200.5,388.2728)	4.88
2	(203.4712,388.7125)	(201.5,388.9712)	5.19
3	(218.3488,375.2252)	(215.3025,375.895)	4.65
4	(223.9468,391.9682)	(225.6064,391.4468)	5.43
5	(227.3262,381.7578)	(228.8984,381.1016)	5.00
6	(231.2535,371.1436)	(231.9965,370.7553)	4.87
7	(236.1746,383.6634)	(233.8603,385.1397)	5.01
8	(246.4136,377.4045)	(247.2037,376.7963)	5.10
9	(249.1863,369.2876)	(252.3137,366.415)	4.86
10	(259.1863,366.2876)	(256.3137,369.415)	4.86

Table 2.1 Motion centers and rotating angles

Time	1702	1708	1714	1720	1802	1808	1814
Rotating angle	9.087668	9.643614	14.33013	6.851278	11.9749	17.34119	10.70173
Time	1814	1820	1902	1908	1914	1920	2002
Rotating angle	10.70173	9.476946	2.7636	19.95497	9.770217	7.0175	5.788656

Table 2.2 Rotating angle for storm Bill.

CHAPTER 3

3-BRANCH HIERARCHICAL CLUSTERING

3.1 Introduction to Hierarchical Clustering

Hierarchical clustering is a method to sort a set of points into groups that each group is considered to have similar points. To achieve that, a hierarchical tree is built with all the points as leaves, and points that are similar with each other have the same root. The current methods mostly build two-branch hierarchical trees, where each root has two branches. Bottom-up technique is the most common way to implement the algorithm. We start with two closest points, create a cluster by combining these two points. Then, by substituting the new cluster with one point, the set of points are modified and there is one point less than the original set. The process goes on until we only have one point left in the set. There are a couple of questions in the building process. The first question is how to represent the new cluster generated by two close points. One solution for the question is to use the midpoint as a representative, whose coordinates are defined by the average of coordinates of the points. As we will see in later discussion, when dealing with multiple points, the position of midpoints tend be sensitive to noises. Another question with the two-branch hierarchical tree is that the system itself is not stable when adding or removing points from the set. In this chapter, we design a new algorithm about how to represent a group of points with one point and apply it to the construction of alternative hierarchical trees.

3.2 Fermat Point

In this section, we review the previous methods of representing a group of

points with one point and introduce a new candidate. The new point has the property of being stable when disturbed by perturbation and including information from every point in the set.

3.2.1 Two ways of representing a set of points

When dealing with a set of points, there are situations where only one point is required to summarize the characteristics of the set. How to identify the point from the set is an interesting problem in data analysis. Because every point in the set should have the same degree of importance, we could not use any one particular point from the set. Instead, we need to define a point which is contributed by every other point in the set. There are generally two popular ways to define the representing point: Mean and Median.

Definition: Given a set of points $a_1, a_2, a_3, \dots, a_m \in \mathbb{R}^n$, their corresponded coordinates are $(x_{11}, x_{12}, x_{13}, \dots, x_{1n})$, $(x_{21}, x_{22}, x_{23}, \dots, x_{2n})$, $(x_{31}, x_{32}, x_{33}, \dots, x_{3n})$, $\dots$, $(x_{m1}, x_{m2}, x_{m3}, \dots, x_{mn})$. The Mean Point of the set is defined by:

$$x_0 = \left(\frac{x_{11} + x_{21} + x_{31} + \cdots + x_{m1}}{m}, \frac{x_{12} + x_{22} + x_{32} + \cdots + x_{m2}}{m}, \right.$$
$$\left. \frac{x_{13} + x_{23} + x_{33} + \cdots + x_{m3}}{m}, \dots, \frac{x_{1n} + x_{2n} + x_{3n} + \cdots + x_{mn}}{m}\right) \quad (3.1)$$

Definition: Given a set of points $a_1, a_2, a_3, \dots, a_m \in \mathbb{R}^n$, their corresponded coordinates are $(x_{11}, x_{12}, x_{13}, \dots, x_{1n})$, $(x_{21}, x_{22}, x_{23}, \dots, x_{2n})$, $(x_{31}, x_{32}, x_{33}, \dots, x_{3n})$, $\dots$, $(x_{m1}, x_{m2}, x_{m3}, \dots, x_{mn})$. The kth coordinate of the Median Point is defined by the

median of the sequence $\{x_{1k},\ x_{2k},\ x_{3k}, \dots, x_{mk}\}$, $k = 1,2,3, \dots, n$.

We will prove that the Mean Point of the set minimizes of summation of L^2 distance, which means that from the Mean Point, when we add all the distance between Mean Point and points of the set in L^2 form, we have the smallest summation. The mathematical proof is in the following section.

The Euclidean distance between two n-dimension points $a_1\ (x_{11}, x_{12}, x_{13}, \dots, x_{1n})$, $a_2(x_{21}, x_{22}, x_{23}, \dots, x_{2n})$ is defined by:

$$d(a_1, a_2)$$

$$= \sqrt{(x_{11} - x_{21})^2 + (x_{12} - x_{22})^2 + (x_{13} - x_{23})^2 + \cdots + (x_{1n} - x_{2n})^2}. \quad (3.2)$$

From $a_o\ (x_1, x_2, x_3, \dots, x_n) \in \mathbb{R}^n$, the L^2 total distance for the set A is:

$$f(a_0, A) = \sqrt{\sum_{i=1}^{m} d(a_0, a_i)^2}$$

If $f(a_0, A)$ is the minimum, we have:

$$\frac{\partial f}{\partial x_k}(a_0, A) = 0, k = 1,2,3, \dots, n.$$

Which is:

$$\frac{\partial}{\partial x_k}\sqrt{\sum_{i=1}^{m} d(a_0, a_i)^2} = 0, \qquad k = 1,2,3, \dots, n.$$

$$\frac{1}{2\sqrt{\sum_{i=1}^{m} d(a_0, a_i)^2}}\frac{\partial}{\partial x_k}\sum_{i=1}^{m} d(a_0, a_i)^2 = 0, \qquad k = 1,2,3, \dots, n.$$

$$\sum_{i=1}^{m}\frac{\partial}{\partial x_k} d(a_0, a_i)^2 = 0, \qquad k = 1,2,3, \dots, n.$$

$$\sum_{i=1}^{m} \frac{\partial}{\partial x_k} \sum_{j=1}^{n} (x_j - x_{ij})^2 = 0, \qquad k = 1,2,3, \dots, n.$$

$$\sum_{i=1}^{m} 2(x_k - x_{ik}) = 0, \qquad k = 1,2,3, \dots, n.$$

Therefore, by solving for x_k, $k = 1,2,3, \dots, n$, we have:

$$x_k = \frac{x_{k1} + x_{k2} + x_{k3} + \cdots + x_{km}}{m}, \quad k = 1,2,3, \dots, n. \qquad (3.3)$$

That is the consistent with the definition of Mean Point and proves that Mean Point minimizes the L^2 summation.

From the definition of Median Point, we could see that every coordinate of the Median Point only depends on the corresponded coordinate of points in the set. It lacks the ability of treating the points as a whole system, instead, the points are split into independent parts. Therefore, the Median Point is hardly convincing to be a good candidate to represent the whole set.

3.2.2 Fermat Point

It could be seen that every coordinate of the points appears directly in the formula of the Mean Point, that means noises could affect the position of Mean Point straightforwardly. To decrease the disturbance of noises, we modify the definition of Mean Point by using L^1 summation instead, to get the Fermat Point. Now we try to find a point that minimizes the L^1 total distance of set A, which is:

$$g(a_0, A) = \sum_{i=1}^{m} d(a_0, a_i).$$

We start from partial derivative of g with respect to x_k:

$$\frac{\partial g}{\partial x_k}(a_0, A) = 0, k = 1,2,3, \dots, n.$$

$$\frac{\partial}{\partial x_k}\sum_{i=1}^{m} d(a_0, a_i) = 0, \qquad k = 1,2,3, \dots, n.$$

$$\sum_{i=1}^{m} \frac{\partial}{\partial x_k}\sqrt{\sum_{j=1}^{n}(x_j - x_{ij})^2} = 0, \qquad k = 1,2,3, \dots, n.$$

$$\sum_{i=1}^{m} \frac{x_k - x_{ik}}{\sqrt{\sum_{j=1}^{n}(x_j - x_{ij})^2}} = 0, \qquad k = 1,2,3, \dots, n.$$

$$\sum_{i=1}^{m} \frac{x_k}{\sqrt{\sum_{j=1}^{n}(x_j - x_{ij})^2}} = \sum_{i=1}^{m} \frac{x_{ik}}{\sqrt{\sum_{j=1}^{n}(x_j - x_{ij})^2}}, \qquad k = 1,2,3, \dots, n.$$

$$x_k = \frac{\sum_{i=1}^{m} \frac{x_{ik}}{\sqrt{\sum_{j=1}^{n}(x_j - x_{ij})^2}}}{\sum_{i=1}^{m} \frac{1}{\sqrt{\sum_{j=1}^{n}(x_j - x_{ij})^2}}}, \quad k = 1,2,3, \dots, n. \tag{3.4}$$

Definition: Given a set of points $a_1, a_2, a_3, \dots, a_m \in \mathbb{R}^n$, their corresponded coordinates are $(x_{11}, x_{12}, x_{13}, \dots, x_{1n})$, $(x_{21}, x_{22}, x_{23}, \dots, x_{2n})$, $(x_{31}, x_{32}, x_{33}, \dots, x_{3n})$, … , $(x_{m1}, x_{m2}, x_{m3}, \dots, x_{mn})$. The Fermat Point of the set is defined by:

$$a_0 = \left(\frac{\sum_{i=1}^{m} \frac{x_{i1}}{\sqrt{\sum_{j=1}^{n}(x_j - x_{ij})^2}}}{\sum_{i=1}^{m} \frac{1}{\sqrt{\sum_{j=1}^{n}(x_j - x_{ij})^2}}}, \frac{\sum_{i=1}^{m} \frac{x_{i2}}{\sqrt{\sum_{j=1}^{n}(x_j - x_{ij})^2}}}{\sum_{i=1}^{m} \frac{1}{\sqrt{\sum_{j=1}^{n}(x_j - x_{ij})^2}}}, \right.$$

$$\left(\frac{\sum_{i=1}^{m} \frac{x_{i3}}{\sqrt{\sum_{j=1}^{n}(x_j - x_{ij})^2}}}{\sum_{i=1}^{m} \frac{1}{\sqrt{\sum_{j=1}^{n}(x_j - x_{ij})^2}}}, \ldots, \frac{\sum_{i=1}^{m} \frac{x_{in}}{\sqrt{\sum_{j=1}^{n}(x_j - x_{ij})^2}}}{\sum_{i=1}^{m} \frac{1}{\sqrt{\sum_{j=1}^{n}(x_j - x_{ij})^2}}} \right) \tag{3.5}$$

We use iteration to find the Fermat Point from the above definition. From the definition above it could be seen that the denominators might be zeros during the iteration process. To avoid the situation of having a zero denominator, in the process of iteration, a small modification is made by adding a nonzero positive number ε. The following formula is used:

$$a_0 = \left(\frac{\sum_{i=1}^{m} \frac{x_{i1}}{\sqrt{\sum_{j=1}^{n}(x_j - x_{ij})^2 + \varepsilon}}}{\sum_{i=1}^{m} \frac{1}{\sqrt{\sum_{j=1}^{n}(x_j - x_{ij})^2 + \varepsilon}}}, \frac{\sum_{i=1}^{m} \frac{x_{i2}}{\sqrt{\sum_{j=1}^{n}(x_j - x_{ij})^2 + \varepsilon}}}{\sum_{i=1}^{m} \frac{1}{\sqrt{\sum_{j=1}^{n}(x_j - x_{ij})^2 + \varepsilon}}}, \right.$$

$$\left. \frac{\sum_{i=1}^{m} \frac{x_{i3}}{\sqrt{\sum_{j=1}^{n}(x_j - x_{ij})^2 + \varepsilon}}}{\sum_{i=1}^{m} \frac{1}{\sqrt{\sum_{j=1}^{n}(x_j - x_{ij})^2 + \varepsilon}}}, \ldots, \frac{\sum_{i=1}^{m} \frac{x_{in}}{\sqrt{\sum_{j=1}^{n}(x_j - x_{ij})^2 + \varepsilon}}}{\sum_{i=1}^{m} \frac{1}{\sqrt{\sum_{j=1}^{n}(x_j - x_{ij})^2 + \varepsilon}}} \right) \tag{3.6}$$

The Mean Point is a good candidate as the initial value for a_0. The convergence of the iteration to find a_0 is proven by Weiszfield (Weiszfeld, E., 1937), Kuhn(Kuhn, H, 1973) and Vardi and Zhang (Vardi, Y., Zhang, 2000).

In Section 3.4, through experiments, we will find that the Fermat Point is more consistent than the Mean Point when dealing with disturbance, and noises will now affect the position of Fermat Point to a insignificant degree. Thus, the Fermat Point could be used to be a efficient candidate to represent a group of points. Using the Fermat Point, we develop a new technique for hierarchical clustering, as shown in

Section 3.3.

3.3 3-branch Hierarchical Tree

In the creation of an 2-branch hierarchical tree for a set of points, we need to combine two points and generate a new point which will represent the existing two. The Mean Point is used in the case of creating 2-branch hierarchical tree. Since the Mean Point is easily disturbed by noises, when creating 3-branch hierarchical trees, we need a new way to represent three points. The Fermat Point is thus introduced in this section. Through the experiment in Section 3.4, we find that the Fermat Point is more stable than the Mean Point when dealing with noises.

For a given set of points, to construct a 3-branch hierarchical tree, we find the group of three points that has the smallest distance, and use the Fermat Point of the group as a new point to replace those three points. The distance we use here is the summation of the distances between the Fermat Point and each point in the group. In each step, we represent three points using one point, by reiterating the process, the number of points in the set is decreasing by 2 every time. The iteration stops when there are less than four points left in the set, as shown in Figure 3.1.

3.4 Experiments

3.4.1 Mean Point and Fermat Point

In this experiment, we use a set with three points as an example. The points in the set form a close group, as shown in Figure 3.1. The difference between the Mean Point and Fermat Point is relatively small. Both are good candidates to represent the

group. After an outlier point is added to the group far on the top, the Mean Point and Fermat Point are calculated again and the results are shown in Figure 3.2. It could be seen clearly that the Mean Point is affected significantly while the Fermat Point stays at the same spot. Now, to use the Mean point as a representative for the group seems much less convincing, because it consists of too much information from one outlier and ignores the impact of the majority of the set. Meanwhile, Fermat Point remains a good candidate when noises are disturbing the data.

3.4.2 3-Branch Hierarchical Tree

In this section, the newly devised algorithm is put into application. We construct a 2-branch hierarchical tree and a 3-branch hierarchical tree for a set of 20 points. In Figure 3.3, we display the points on the 2-dimension plane. Figure 3.4 shows the 2-branch hierarchical tree and Figure 3.5 shows the 3-branch hierarchical tree for the set. By comparing the new 3-branch diagram with the 2-branch tree, we could see that it is much easier to distinguish the clusters of points in the 3-branch tree.

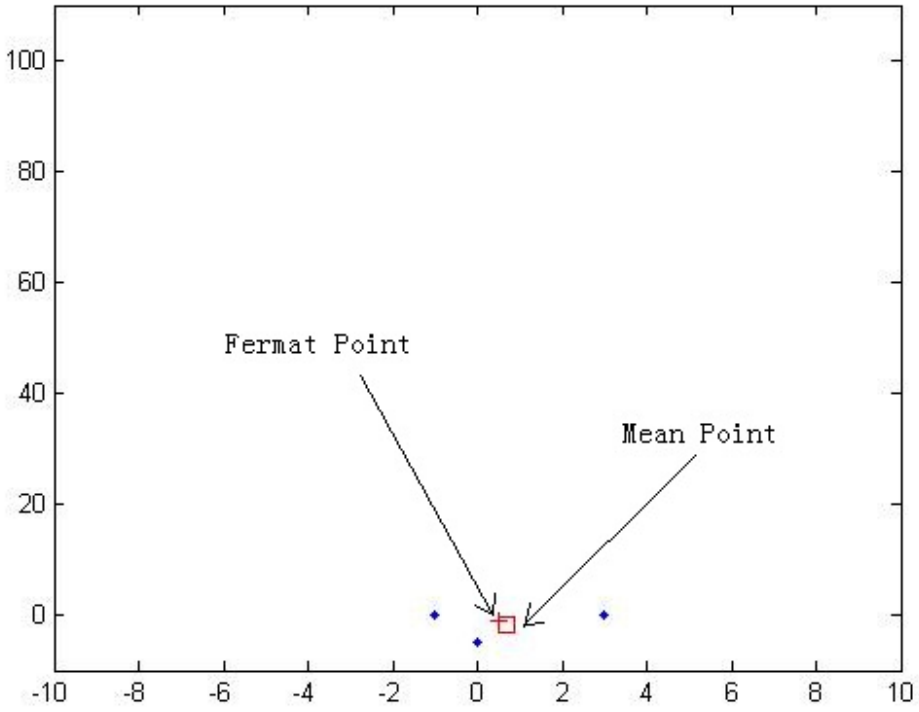


Figure 3.1 Mean Point and Fermat Point of three points

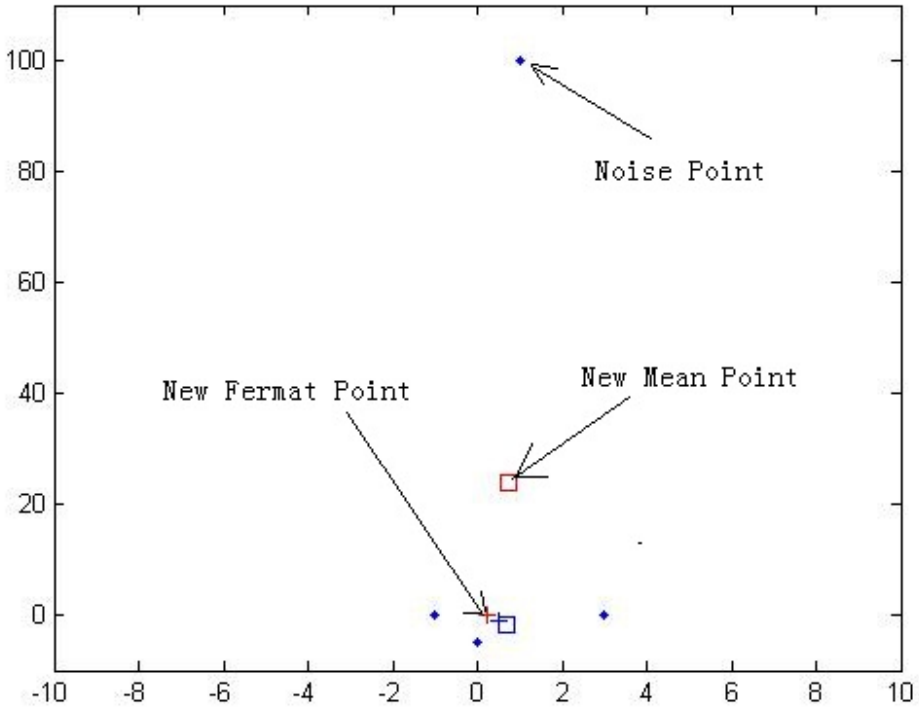


Figure 3.2 The disturbance of a noise point on Mean Point and Fermat Point

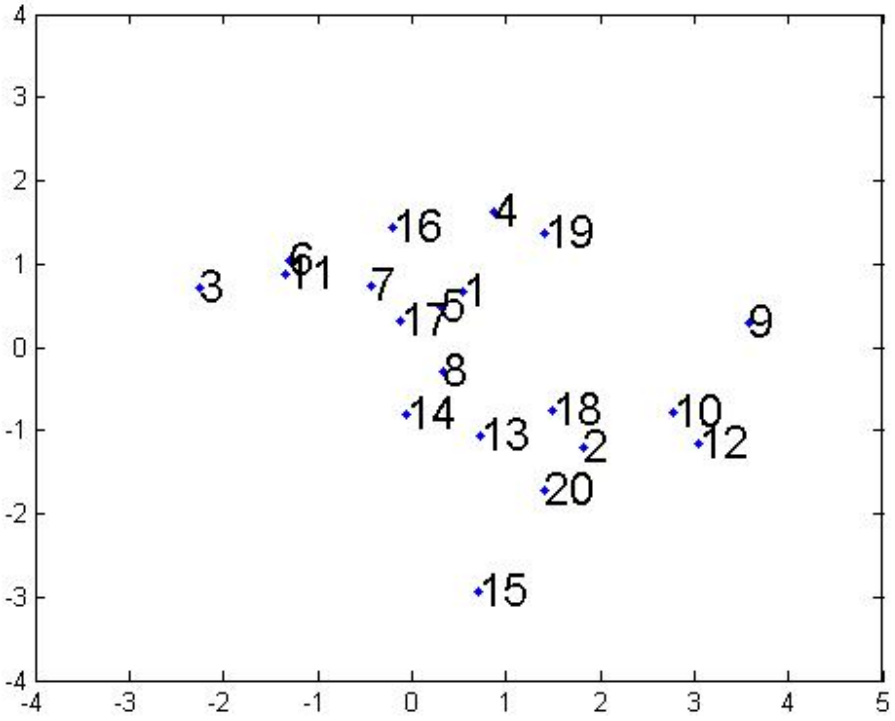


Figure 3.3 20 points on the plane

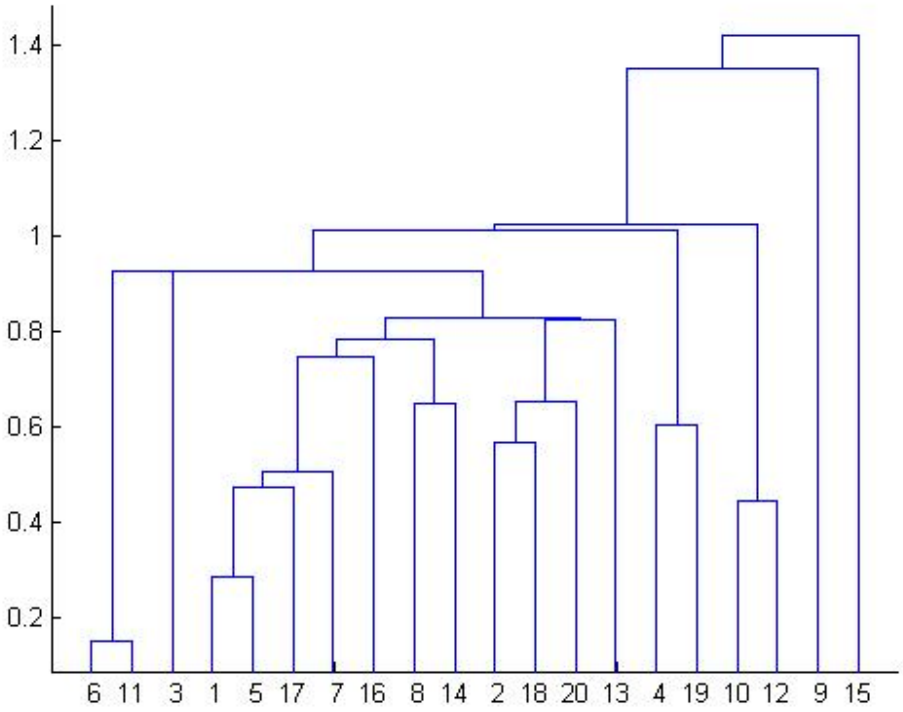


Figure 3.4 The 2-branch diagram

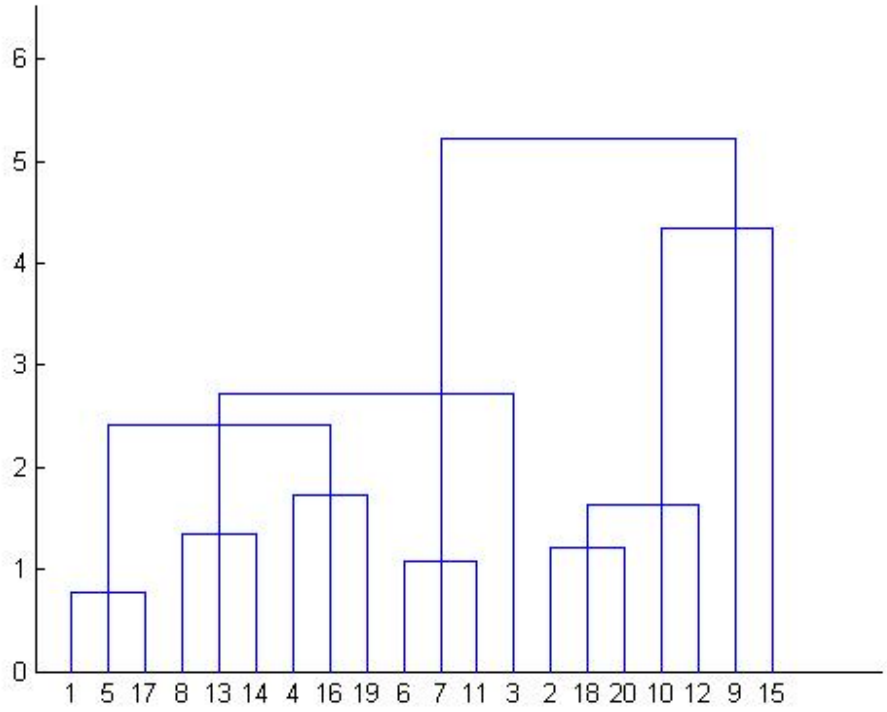


Figure 3.5 The 3-branch diagram

BIBLIOGRAPHY

Ahijevych, D., Gilleland, E., Brown, B., Ebert, E., 2009. Application of spatial verification methods to idealized and NWP gridded precipitation forecasts. *Weather and Forecasting* 24, 1485 – 1497.

Alexander, G., Weinman, J., Karyampudi, V., Olson, W., Lee, A., 1999. The effect of assimilating rain rates derived from satellites and lightning on forecasts of the 1993 superstorm. *Mon. Wea. Rev.* 127, 1433 – 1457.

Baddeley, A.J., 1992. An errormetric for binary images. In: Förstner,W., Ruwiedel, S. (Eds.), *Robust Computer Vision: Quality of Vision Algorithms.Wichmann*, pp. 59 – 78.

Baldwin, M., Mitchell, K., 1998. Progress on the NCEP hourly multi-sensor U.S. precipitation analysis for operations and GCIP research. 2nd Symp. on Integrated Observing Systems. *Amer. Meteor. Soc*., Phoenix, AZ, pp. 10 – 11.

Caselles, V, R. Kimmel and G. Sapiro, "Geodesic active contours," *International journal of computer vision,* vol. 22, pp. 61-79, 1997.

Casati, B., Wilson, L., 2007. A new spatial-scale decomposition of the Brier score: application to the verification to lightning probability forecasts. *Monthly Weather Review* 135 (9), 3052 – 3069.

Chan, T and L. Vese, "Active contours without edges," *IEEE Transactions on image processing,* vol. 10, pp. 266-277, 2001.

Cuisenaire, O., Macq, B., 1999. Fast Euclidean distance transformation by propagation using multiple neighborhoods. *Computer Vision and Image Understanding* 76 (2), 163 – 172.

Davis, C., Brown, B., Bullock, R., 2006. Object-based verification of precipitation forecasts. Part i: methodology and application to mesoscale rain areas. *Monthly Weather Review* 134 (7), 1772 – 1784.

Dixon, M and G. Wiener, ``TITAN: Thunderstorm identification, tracking, analysis, and nowcasting--A radar-based methodology," *Journal of Atmospheric and Oceanic Technology,* vol. 10, pp. 785-797, 1993.

Ebert, E., 2009. Neighborhood verification: a strategy for rewarding close forecasts. *Weather and Forecasting* 24 (6), 1498 – 1510.

Fabbri, R., Costa, L., Torelli, J., Bruno, O., 2008. 2D Euclidean distance transforms: a comparative survey. *ACM Computing Surveys* 40 (1), 44. doi:10.1145/1322432.1322434 http://doi.acm.org/10.1145/1322432.1322434.

Gilleland, E., Lee, T., Gotway, J., Bullock, R., Brown, B., 2008. Computationally efficient spatial forecast verification using Baddeley's delta image metric. *Monthly Weather Review* 136 (5), 1747 – 1757.

Gilleland, E., Ahijevych, D., Brown, B.G., Casati, B., Ebert, E.E., 2009. Intercomparison of spatial forecast verification methods. *Weather and Forecasting* 24 (5), 1416 – 1430.

Janjic, Z., Black, T., Pyle, M., CHuang, H., Rogers, E., DiMego, G., 2005. High resolution applications of the WRF NMM. 21st Conf. on Weather Analysis and Forecasting/17th Conf. on Numerical Weather Prediction. *Amer. Meteor. Soc.*, Washington, DC. p. 16A.4.

Johnson, J et al., "The storm cell identification and tracking algorithm: An enhanced WSR-88D algorithm," *Weather and Forecasting*, vol. 13, pp. 263-276, 1998.

Kain, J.S., Weiss, S.J., Bright, D.R., Baldwin, M.E., Levit, J.J., Carbin, G.W., Schwartz, C.S., Weisman, M.L., Droegemeier, K.K., Weber, D.B., Thomas, K.W., 2008. Some practical considerations regarding horizontal resolution in the first generation of operational convection-allowing NWP. *Weather and Forecasting* 23 (5), 931 – 952.

Kass, M, A. Witkin and D. Terzopoulos, "Snakes: Active contour models," *International journal of computer vision*, vol. 1, pp. 321-331, 1988.

Keil, C., Craig, G., 2009. A displacement and amplitude score employing an optical flow technique. *Weather and Forecasting* 24, 1297 – 1308.

Kuhn, H. A note on Fermat's problem, *Mathematical Programming*, Volume: 4 Issue: 1(1973), 98-107.MAT'S

Lakshmanan, *et al.*, "Multiscale storm identification and forecast," *Atmospheric Research*, vol. 67, pp. 367-380, 2003.

Lakshmanan and T. Smith, "Data mining storm attributes from spatial grids," *J. Ocea. and Atmos. Tech*, 2008.

Lakshmanan, V., Kain, J., 2010. A Gaussian mixture model approach to forecast verification. *Weather and Forecasting* 25 (3), 908 – 920.

Marzban, C., Sandgathe, S., 2006. Cluster analysis for verification of precipitation

fields. *Weather and Forecasting* 21 (5), 824 – 838.

Mumford, D and J. Shah, "Optimal approximations by piecewise smooth functions and associated variational problems," *Comm. Pure Appl. Math*, vol. 42, pp. 577-685, 1989.

Osher, S and Sethian, J, "Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations," *Journal of computational physics*, vol. 79, pp. 12-49, 1988.

Paragios, N Y. Chen and O. Faugeras, *Handbook of mathematical models in computer vision*: Springer-Verlag New York Inc, 2006.

Rogers, E., DiMego, G., Black, T., Ek, M., Ferrier, B., Gayno, G., Janjic, Z., Lin, Y., Pyle, M.,Wong, V.,Wu,W., Carley, J., 2009. The NCEP North American meso scale modeling system: recent changes and future plans. 23rd Conf. on Weather Analysis and Forecasting/19th Conf. on Numerical Weather Prediction. *Amer. Meteor. Soc.*, Omaha, NE. p. 2A.4.

Rucklide, W., 1996. Effective Visual Recognition Using the Hausdorff Distance. *Springer*, New York.

Saito, T., Toriwaki, J.-I., 1994. New algorithms for Euclidean distance transformation of an n-dimensional digitized picture with applications. *Pattern Recognition* 27 (11), 1551 – 1565.

Skamarock, W., Klemp, J., Dudhia, J., Gill, D., Barker, D., Wang, W., Powers, J., 2005. A description of the Advanced Research WRF version 2. Tech. Rep. NCAR/TN-468*STR. National Center for Atmospheric Research, Boulder, CO, Available from *UCAR Communications*, P.O. Box 3000, Boulder CO 80307.

Tuttle, J and R. Gall, "A single-radar technique for estimating the winds in tropical cyclones," *Bulletin of the American Meteorological Society*, vol. 80, 1999.

Vardi, Y., Zhang, C-H. The multivariate L1-median and associated data depth, *Proceedings Of The National Academy Of Sciences Of The United States Of America* (PNAS) Volume: 97 Issue: 4 (2000), 1423-1426.

Vardi, Y., Zhang, C-H. A modified Weiszfeld algorithm for the Fermat-Weber location problem, *Mathematical Programming*, Volume: 90 Issue: 3 (2001), 559-566.

Venugopal, V., Basu, S., Foufoula-Georgiou, E., 2005. A new metric for comparing precipitation patterns with an application to ensemble forecasts. *J. Geophysical Research* 110 (D08111).

Rudin, W, Real and Complex Analysis, Third Edition.

Wernli,H., Hofmann, C., Zimmer, M., 2009. Spatial forecast verification methods inter-comparison project — application of the SAL technique. *Weather and Forecasting* 24. doi:10.1175/2009WAF2222271.1.

Weiszfeld, E. (1937): Sur le point pour lequel la somme des distances de n points donnés est minimum. *Tôhoku Mathematics Journa,l* 43, 355–386.

Zhang, P and M. Zhu, "Radar Image Segmentation Using Active Contour Method," *34th Conference on Radar Meteorology*, Williamsburg, VA, USA, AMS, CD-ROM, P9.2..

Zhu, M and P. Zhang, "Segmentation for radar images based on active contour," *preprint*, 2009.